Analysis of Machine Learning Techniques for Database Optimization

Darshit Amit Pandya

Northwestern University, USA

Corresponding author: Darshit Amit Pandya, Email: darshitpadya211@gmail.com

This study discusses the available database optimization techniques to overcome the limitations of the traditional (empirical) database optimization techniques by analyzing and leveraging the machine learning techniques to meet the high performance demands and diverse workloads of today's world, by exploring and analyzing several categories of possible improvements, including query optimization, optimizing indexing strategies, reducing resource utilization, optimizing entity matching, and enhancing cardinality estimation. Through this technical overview, the survey aims to provide a more comprehensive understanding of the advancements and complexities in leveraging machine learning for enhancing database performance and efficiency. This study also explores the limitations and uncovered areas of the existing SOTA, and further explores the recent advancements and novel approaches proposed in the domain of intelligent database optimization, and how they overcome the limitations of SOTA techniques. Finally, certain problem statements are deduced and listed to guide future research in the domain of intelligent database optimization.

Keywords: Machine Learning, Deep Learning, Databases, Optimization, SQL.

2024. In Mukesh Saraswat & Rajani Kumari (eds.), *Applied Intelligence and Computing*, 73–80. Computing & Intelligent Systems, SCRS, India. DOI: https://doi.org/10.56155/978-81-955020-9-7-9

1 Introduction

The 21st century is said to be the century of Data. As is evident from the technological advancements, data serves as the driving force behind the fast-paced growth in businesses and economies, or for that matter, the whole of human life. It is safe to say that data is the new oil. Due to the increasing accessibility of technology, the amount of data generated is absolutely extraordinary. Accordingly, it becomes crucial to put in place a robust and efficient data management system to handle such high volumes of data efficiently. There exists a vast pool of Database Management Systems capable of efficiently processing database queries. However, due to the diverse workloads and exponentially growing volumes of data, such systems turn out to be inefficient in managing the database underneath. As we have commenced on AI adventure, it makes sense to leverage the power of AI to increase the efficiency of the database management systems.

This paper identifies and outlines the aspects of the database management systems that could potentially benefit from AI. Specifically, the paper discusses the traditional (empirical) approaches toward database optimization, followed by the newer, more advanced machine learning and deep learning techniques that have been proposed in the recent past.

2 Motivation

In the rapidly evolving landscape of database management techniques, optimization of performance and efficiency have been increasingly challenging. Especially, the quality, quantity, and accessibility of data are crucial for the performance of machine learning and deep learning algorithms that powers many systems in today's world and continues to do so. However, to increase the performance, the quantity of data required also increases which leads to a major challenge in the data management aspect as it becomes difficult to handle and manage such a large amount of data.

There are plentiful articles and techniques available for improving the training data characteristics using various machine learning techniques that leads to substantial performance improvement in the model. All of these techniques refer to the optimization of training data. However, the main focus of this paper is to survey and analyze the machine learning techniques that can be utilized to optimize the database functionality. This could lead to better data accessibility and improved performance of the model which uses this database. There are traditional techniques (e.g. cost estimation, join order selection, knob tuning, index and view advisor) available but they cannot meet the high-performance requirements for large- scale databases having diversified users, like in the cloud. Accordingly, machine learning techniques can help address this issue by judiciously selecting the perfect optimization strategy based on the use case requirements. Some of the major advantages of database optimization include improved query performance, enhanced data retrieval, enhanced scalability, and adaptability to diverse workloads.

3 Databases: Scope of Optimization

Database Management Systems (DMS) are complex systems that look over numerous tasks like Data Storage, Data Retrieval, Data Manipulation, Data Security, Data Integrity, Concurrency Control, Backup and Recovery, Query Optimization, and Data Dictionary Management. Although it might not be possible to optimize all these database tasks, there are certain tasks that could be optimized by utilizing certain techniques for database optimization. The most common tasks that could be optimized, as detailed by Li, et.al. [1], are

- **Knob Space Exploration:** Tuning database configuration parameters (knobs) to optimize performance and resource usage.

- **Index/View Selection:** Choosing the best indexes or materialized views to enhance query performance based on usage patterns.
- **Partition-Key Recommendation:** Selecting the partition key for distributed databases to improve data distribution and query processing.
- **Cardinality Estimation:** Estimating the number of rows returned by queries to aid in selecting the best execution plan.
- **Index Benefit Estimation:** Estimating the performance improvement from creating indexes on specific columns.
- **Query Latency Prediction:** Predicting query execution times based on historical data and query characteristics.
- **Query Workload Prediction:** Forecasting future query workloads to optimize database systems preemptively.

4 Traditional Approaches

There are many existing traditional methods/techniques that have been utilized by practitioners to optimize the database management systems since quite a while. These traditional approaches do not use intelligence or automation (as in AI); Rather, they are logic-based approaches that focus on utilizing logic behind theoretical approaches to optimize the database systems. Some of these approaches include

- **Knob Space Exploration:** Grid search, which systematically explores a predefined set of configurations, can be used to find the optimal configuration by evaluating each point in the grid.
- **Index/View Selection:** Histograms and sampling techniques can be used to estimate the selectivity of different indexes or materialized views based on data distribution statistics.
- **Partition-Key recommendation:** Statistical analysis, such as mean, median, and variance calculations, can be used to identify a partition key that balances data distribution across partitions.
- **Query Rewrite:** Heuristic-based approaches, like rule- based transformations or query simplification techniques, can be used to rewrite queries based on predefined rules.
- **Join Order Selection:** Greedy algorithms, such as the left- deep join tree algorithm, can be used to iteratively select join orders based on estimated join costs.
- **Cardinality Estimation:** Sampling techniques, like random sampling or stratified sampling, can be used to estimate the cardinality of query results based on sampled data.
- **Index Benefit Estimation:** Cost models, based on statistical analysis of query execution plans, can be used to estimate the benefit of creating an index based on query patterns.
- **Query Latency Prediction:** Time series analysis techniques, such as exponential smoothing or moving averages, can be used to predict query execution times based on historical latency data.
- **Query Workload Prediction:** Markov models or autoregressive models can be used to predict future query workloads based on past query sequences and their frequencies.

5 State of the Art (SOTA) Techniques

Each of the traditional approaches were quite widely-used when they were proposed (in the late-90s and early-2000s). However, as is with research, it gets better with technological advancements. Accordingly, with the arrival of AI, these tasks can be improved by incorporating machine learning and deep learning algorithms. Some of the high performance techniques covered in the comparative summary by Guoliang Li, et.al. [1] include

- Knob Space Exploration: Gradient-based [2]
- Index/View Selection: q-learning [3]
- Partition-Key recommendation: q-learning [4]
- Query Rewrite: MCTS [5]
- Join Order Selection: q-learning [6]
- Cardinality Estimation: Dense Network [7]
- Index Benefit Estimation: Dense Network [8]
- Query Latency Prediction: Graph Embedding [9]
- Query Workload Prediction: q-learning [10]

5.1 Limitations of SOTAs

Although these techniques do perform well, there are certain challenges [1] as well including Model Selection, Training Data, Adaptability, and Model Convergence. Based on the performance, these approaches do not meet the desired level of automation and can be improved with newer, more robust, generalizable methods in the following areas

- QLearned Entity Matching
- Learned Cardinality Estimation
- Learned Query Optimization
- Learned Index Structures
- Adaptability for diverse Workloads
- Learning for OLAP and TLAP, Sorting algorithms, and Query executor

Accordingly, it is quite clear that there exists certain aspects of the database systems that can be optimized with more advanced deep learning techniques to meet the needs.

5.2 Novel Approaches and Research Proposals

In this section, an overview of the work accomplished in the above-mentioned areas is studied to get an idea as to how the level of current research overcomes the limitations of lacking automations in the said areas.

5.3 Technological Leap

The technological leap in database optimization has progressed from basic indexing techniques like B-trees and hash indexes to the development of relational database systems (RDBMS) and standardized query languages like SQL, enabling more efficient data organization and retrieval. With the rise of big data and cloud computing, distributed database systems such as Hadoop and Spark have emerged to address scalability challenges. The introduction of machine learning and deep learning has further revolutionized database optimization, with techniques like learned indexes leveraging deep learning models to improve query performance and resource utilization. This evolution highlights a shift towards intelligent, adaptive database systems capable of handling the complexities of modern data management. With the dawn of Generative AI and advanced deep learning techniques, coupled with high-performance hardware (GPUs), optimization can become more intelligent.

1) **Learned Entity Matching:** The Ditto entity matching system [11] is based on pretrained Transformer-based language models and fine-tunes them for entity matching (EM) as a sequence-pair classification problem. It significantly improves matching quality over previous state-of-the-art (SOTA) methods, achieving up to a 29% increase in F1 score on benchmark datasets using models like BERT, DistilBERT, or RoBERTa. Ditto introduces three optimization techniques to enhance matching capability: injecting domain knowledge by highlighting important input information, summarizing long strings to retain essential

information, and adapting a SOTA data augmentation technique for text to enhance training data with difficult examples. These optimizations boost Ditto's performance by up to 9.8%. Surprisingly, Ditto achieves previous SOTA results with only half the labeled data. In a real-world large-scale EM task matching two company datasets with 789K and 412K records, Ditto achieves a high F1 score of 96.5%, demonstrating its effectiveness.

2) **Learned Cardinality Estimation:** PostCENN [12] introduces an innovative approach in PostgreSQL by integrating machine learning models for cardinality estimation, aiming to enhance accuracy for specific database schema segments. Unlike traditional methods such as histograms and sampling, which can be limited by simplifying assumptions, PostCENN leverages neural networks (NNs) to provide more precise estimates, especially in complex data scenarios. This integration follows a structured lifecycle, including model creation, training, and transparent use within the query optimizer, offering a novel solution to the challenges of cardinality estimation in modern database systems.
3) **Learned Query Optimization:** Bao [13] introduces a bandit optimizer that enhances query optimization using reinforcement learning. Unlike previous efforts that faced challenges like high training overhead and inability to adapt, Bao leverages modern tree convolutional neural networks and Thompson sampling to automatically learn and adapt to changes in query workloads, data, and schema. Experimental results demonstrate that Bao can quickly learn strategies to improve query execution performance, including reducing tail latency, and can offer cost and performance benefits compared to commercial systems in cloud environments.
4) **Learned Index Structures:** The paper on Learned Index Structures [14] explores the concept of learned indexes, which suggests that traditional index structures like B-Tree, Hash, and BitMap indexes can be replaced with learned models, including deep-learning models. The authors theoretically analyze when learned indexes might outperform traditional indexes and discuss the challenges in designing them. Initial results indicate that learned indexes could offer significant advantages over traditional ones, hinting at the potential for future system designs. The research demonstrates that learned indexes can be beneficial by leveraging the data distribution being indexed, prompting further exploration into various machine learning model types and their combinations with traditional data structures. A particularly intriguing research direction is extending learned indexes to multi-dimensional indexes, where models like neural networks could excel in capturing complex high-dimensional relationships. Moreover, learned algorithms could potentially enhance sorting and joins, not just indexing. While GPU/TPU invocation latency remains a challenge, techniques like batching requests can mitigate this issue.
5) **Diverse Workloads, OLAP-TLAP, Sorting, and Query Executor:** The authors of SageDB [15] highlight the limitations of current general-purpose data processing systems, which do not fully exploit the characteristics of specific applications and data. They introduce SageDB, a data processing system that specializes in an application through code synthesis and machine learning, learning data distribution, workload, and hardware to optimize access methods and query plans. SageDB's approach differs significantly from traditional database systems, presenting new challenges in databases, machine learning, and programming systems. SageDB aims to automatically synthesize index structures, sorting and join algorithms, and entire query optimizers based on learned data patterns and correlations. The authors argue that machine learning models can replace core components of a database system, such as index structures, sorting algorithms, and the query executor, to improve performance. It discusses the potential benefits of learned indexes for various data patterns, including approximate query processing, predictive modeling, and efficient updates. The paper also addresses complexityanalysis and robustness guarantees, aiming to define complexity classes for learned algorithms and provide worst- case performance guarantees in case of data distribution or workload shifts.

6 Observations

- Advanced Deep Learning models (like Transformers) enable semantic comprehension of input information leading to a significant increase (of about 29%) in the F1-score and enables the model to outperform previous SOTAs by utilizing only half of the labeled data, leading to computational efficiency.
- Deep Learning techniques (like Neural Networks) enable the system to achieve more accurate estimations of cardinality based on specific database schema segments, and leads to a more generalizable and efficient model for more complex modern database systems.
- Reinforcement learning and Tree CNN (CONVOLUTIONAL NEURAL NETWORKS) enable the system to learn and adapt to changes in query workloads, data, and schema, making the system adaptable to diverse workloads.
- Learning data distribution, workload, and hardware technicalities can enable the system to optimize access methods and query plans for a database system, irrespective of the complexity and domain of application. This can help to automatically synthesize index structures, sorting and join algorithms, and entire query optimizers based on learned data patterns and correlations. This advanced learning system improves the generalizability, adaptability, and efficiency of database systems without worrying about the application domain and training data limitations.

7 Limitations

Although the advantages of the recent research proposals seem extraordinary, therestill exist somelimitations which can significantly contribute to the performance of suchsystems. The following are some of the challenges that need to be addressed

- **Model Selection:** As seen earlier, there exists a vast pool of Deep Learningmodels that could potentially benefit the database systems. However, it becomesnecessary to select the most optimal model based upon a particular databaserequirement. This can lead to a processing overhead.
- **Hardware Optimization:** The performance of these advanced deep learningmodels depends upon the efficiency of the computational processing of theunderlying CPU and GPU units. It is still a challenge to optimize the hardwareunits to work efficiently based on the requirements of a specific system.
- **Training Data Dependency:** The efficiency and performance of all of the learnedmodels tend to be based on the quality of the training data being fed.Accordingly, it becomes necessary to collect good quality training data forachieving efficient performance.

8 Future Work

8.1 Opinion

Based on the pros and cons of each of the machine learning-based optimizations, Ibelieve that there isa huge potential for intelligence-based optimizations for thedatabase systems. Although the traditional machine learning and deep learningapproaches can limit the potential generalizability of these optimizations, the newergenerative AI algorithms might prove beneficial in terms of generalizability andadaptability towards diverse workloads. All in all, there's a huge scope of improvementin the optimization techniques, considering the recent technological advancements inthe domain of machine learning and AI.

8.2 Open Research Problems

- **Generative AI:** Explore the use of generative models like GANs or VAEs toenhance database management and optimization. These models can augmentdatasets, improving data scarcity issues, and imbalance, leading to betterdatabase performance. They can also generate optimized query plans, designoptimal index structures, detect anomalies in queries or access patterns,compress data without losing information, and predict the impact of schemachanges.
- **On-line Learning:** Explore the use of online learning for database optimization. Itinvolves using algorithms that continuously adapt database management andoptimization processes based on incoming data and user interactions. Thesealgorithms could adapt query optimization strategies, dynamically adjust indexstructures, predict future query workloads, automatically tune databaseparameters, and detect anomalies in database behavior. By leveraging real-timefeedback and learning from historical data, online learning could enable thedatabase system to optimize resource allocation, query processing strategies,and configuration settings, thereby improving overall performance in dynamicenvironments.
- **Mixture of Experts (MoEs):** Explore the usage of MoE models like FlexMoE[16] that can optimize the computational processing required to efficientlyimplement the deep learning-based optimizations in database systems. MoEscan help accommodate such pre-trained or learned database optimizationtechniques within a limited availability of hardware resources.

9 Conclusion

Overall, this paper focuses on analyzing and highlighting the applications of several machine learning techniques that can be utilized directly over the database to improve the overall performance of the system. All in all, this paper takes into consideration the following categories of improvements: knob space exploration, index selection, partition-key recommendation, query rewrite, join or selection, cardinality estimation, index/view benefit estimation, query latency prediction, query workload prediction, and Entity Matching through various machine learning techniques. All in all, these categories broadly cover the most crucial optimization aspects for a database and can help improve the overall performance of the system. Finally, open research problems are deduced from the limitations of these techniques to guide future research.

References

[1] FGuoliang Li, Xuanhe Zhou, and Lei Cao. 2021. Machine learning for databases.Proc. VLDB Endow. 14, 12(July 2021), 3190–3193.

[2] MayureshKunjir and Shivnath Babu. 2020. Black or White? How to Develop anAutoTuner for Memory-based Analytics. In Proceedings of the 2020 ACMSIGMOD International Conference on Management of Data (SIGMOD '20).Association for Computing Machinery, New York, NY, USA, 1667–1683.

[3] H. Lan, Z. Bao, and Y. Peng. An index advisor using deep reinforcement learning.In CIKM, pages 2105–2108, 2020.

[4] Benjamin Hilprecht, Carsten Binnig, and Uwe Röhm. 2020. Learning a PartitioningAdvisor for Cloud Databases. In Proceedings of the 2020 ACM SIGMODInternational Conference on Management of Data (SIGMOD '20). Association forComputing Machinery, New York, NY, USA, 143–157.

[5] G. Li, X. Zhou, S. Ji, X. Yu, Y. Han, L. Jin, W. Li, T. Wang, and S. Li. opengauss:An autonomous database system. VLDB, 2021.

[6] R. C. Marcus, P. Negi, H. Mao, C. Zhang, M. Alizadeh, T. Kraska, O.Papaemmanouil, and N. Tatbul. Neo: A learned query optimizer. PVLDB,12(11):1705–1718, 2019.

[7] A. Kipf, T. Kipf, B. Radke, V. Leis, P. A. Boncz, and A. Kemper. Learnedcardinalities: Estimating correlated joins with deep learning. In CIDR, 2019.

[8] B. Ding, S. Das, R. Marcus, W. Wu, S. Chaudhuri, and V. R. Narasayya. AI meetsAI: leveraging query executions to improve index recommendations. In SIGMOD,pages 1241–1258, 2019

[9] X. Zhou, J. Sun, G. Li, and J. Feng. Query performance prediction for concurrentqueries using graph embedding. VLDB, 13(9):1416–1428, 2020.

[10] C. Zhang, R. Marcus, A. Kleiman, and O. Papaemmanouil. Buffer pool awarequery scheduling via deep reinforcement learning. CoRR, abs/2007.10568, 2020

[11] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020.Deep entity matching with pre-trained language models. Proc. VLDB Endow. 14, 1(September 2020), 50–60.

[12] Lucas Woltmann, Dominik Olwig, Claudio Hartmann, Dirk Habich, and WolfgangLehner. 2021. PostCENN: postgreSQL with machine learning models forcardinality estimation. Proc. VLDB Endow. 14, 12 (July 2021), 2715–2718.

[13] Ryan Marcus, Parimarjan Negi, Hongzi Mao, NesimeTatbul, Mohammad Alizadeh,and Tim Kraska. 2021. Bao: Making Learned Query Optimization Practical. InProceedings of the 2021 International Conference on Management of Data(SIGMOD '21). Association for Computing Machinery, New York, NY, USA,1275–1288.

[14] Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and NeoklisPolyzotis. 2018.The Case for Learned Index Structures. In Proceedings of the 2018 InternationalConference on Management of Data (SIGMOD '18). Association for ComputingMachinery, New York, NY, USA, 489–504.

[15] Kraska, T; Alizadeh, M; Beutel, A; Chi, EH; Ding, J; Kristo, A; Leclerc, G; Madden,S; Mao, H; Nathan, V. 2019. SageDB: A learned database system. In Proceedingsof the 9th Biennial Conference on Innovative Data Systems Research (CIDR2019). MIT DSpace.

[16] XiaonanNie, Xupeng Miao, Zilong Wang, Zichao Yang, JilongXue, Lingxiao Ma,Gang Cao, and Bin Cui. 2023. FlexMoE: Scaling Large-scale Sparse Pre-trainedModel Training via Dynamic Device Placement. Proc. ACM Manag. Data 1, 1,Article 110 (May 2023), 19 pages.