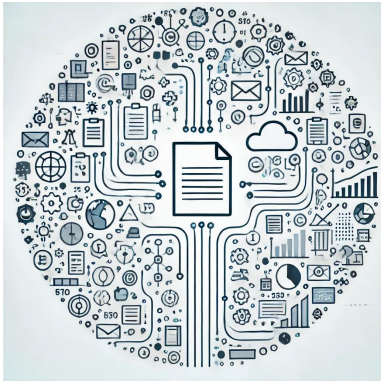
IEEE AIC-2024, Paper-ID: 629

Analysis of Machine Learning techniques for Database Optimization

Darshit Amit Pandya
Northwestern University

Data

- Definition: Raw facts and figures
- Types: Text, numbers, images, etc.
- Sources: Surveys, transactions, sensors



Database

- Purpose: Organizes and stores data efficiently
- Types: Relational (SQL), Non-relational (NoSQL)
- Key Features: Data integrity, security, scalability

Database: Issues

- Slow Query Performance
- High Latency
- Storage Inefficiency
- Concurrency Issues
- High CPU Usage
- Scalability Issues
- Data Redundancy and Inconsistency
- Backup and Recovery Challenges
- Security Vulnerabilities
- Suboptimal Schema Design

Database: Optimization

- **Database Tasks:** Data Storage, Data Retrieval, Data Manipulation, Data Security, Data Integrity, Concurrency Control, Backup and Recovery, Query Optimization, and Data Dictionary Management, etc...
- **Scope of Optimization:**
  - **Knob Space Exploration:** Tuning database parameters for optimal performance.
  - **Index/View Selection:** Choosing best indexes or views for query performance.
  - **Partition-Key Recommendation:** Selecting partition keys for better distribution
  - **Cardinality Estimation:** Estimating rows returned by queries.
  - **Index Benefit Estimation:** Assessing performance gains from new indexes.
  - **Query Latency Prediction:** Predicting query execution times.
  - **Query Workload Prediction:** Forecasting future query workloads

Optimization: Past

- **Knob Space Exploration:** Grid search for optimal configurations.
- **Index/View Selection:** Estimation with histograms and sampling.
- **Partition-Key Recommendation:** Statistical analysis for balanced distribution.
- **Query Rewrite:** Heuristic-based query simplification.
- **Join Order Selection:** Greedy algorithms for optimal join order.
- **Cardinality Estimation:** Sampling techniques for result size estimation.
- **Index Benefit Estimation:** Cost models for index performance benefits.
- **Query Latency Prediction:** Time series analysis for predicting execution times.
- **Query Workload Prediction:** Markov models for future workload prediction.

Logic-based methods: Good Performance, but Not Intelligent/Scalable/Adaptable

Optimization: Present

Machine Learning / Deep Learning-based Methods (Intelligent and Scalable)

- **Knob Space Exploration:** Gradient-based
- **Index/View Selection:** q-learning
- **Partition-Key recommendation:** q-learning
- **Query Rewrite:** MCTS
- **Join Order Selection:** q-learning
- **Cardinality Estimation:** Dense Network
- **Index Benefit Estimation:** Dense Network
- **Query Latency Prediction:** Graph Embedding
- **Query Workload Prediction:** q-learning

Optimization: Present (Limitations)

- **Limitations**
  - Model Selection, Training Data, **Adaptability**, and Model Convergence
  - Does not meet the desired level of automation
- Current Research Standing:
  - Novel, more robust, and generalizable methods in the following areas:
    - Learned 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

Optimization: Key Insights

- **Advanced Deep Learning Models**
  - Transformers improve F1-score by 29% and outperform previous models with half the labeled data, enhancing computational efficiency.
- **Deep Learning Techniques**
  - Neural Networks provide accurate cardinality estimations for complex database systems, leading to more efficient and generalizable models.
- **Reinforcement Learning**
  - These techniques enable systems to adapt to changes in query workloads, data, and schema, making them versatile for diverse workloads.
- **Learning Data Distribution and Workload**
  - Optimizing access methods and query plans based on learned data patterns enhances database systems' generalizability, adaptability, and efficiency, regardless of application domain or training data limitations.

Optimization: Challenges

- **Challenges:**
  - **Model Selection**
    - Choosing the most optimal Deep Learning model based on database requirements
  - **Hardware Optimization**
    - Optimize CPU and GPU units for efficient model performance.
  - **Training Data Dependency**
    - Ensure high-quality training data for effective model performance.

Optimization: Future (Open Research Problems)

- **Generative AI:**
  - Explore generative models
    - Augment datasets to address data scarcity and imbalance
    - Generate optimized query plans
    - Design optimal index structures
    - Detect anomalies in queries or access patterns
    - Compress data without losing information
    - Predict the impact of schema changes
- **Online Learning:**
  - Utilize online learning for continuous database optimization
    - Adapt query optimization strategies based on incoming data
    - Dynamically adjust index structures
    - Predict future query workloads
    - Automatically tune database parameters
    - Detect anomalies in database behavior
    - Optimize resource allocation, query processing strategies, and configuration settings in real-time
- **Mixture-of-Experts (MoEs):**
  - Implement MoE models like FlexMoE for efficient deep learning optimizations
    - Optimize computational processing for database systems
    - Accommodate pre-trained or learned optimization techniques within limited hardware resources

References

1. F Guoliang Li, Xuanhe Zhou, and Lei Cao. 2021. Machine learning for databases. Proc. VLDB Endow. 14, 12 (July 2021), 3190–3193.
2. Mayuresh Kunjir and Shivnath Babu.2020.Black or White? How to Develop an Auto Tuner for Memory-based Analytics. In Proceedings of the 2020 ACM SIGMOD 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 Partitioning Advisor for Cloud Databases. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD '20). Association for Computing 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. open gauss: 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. Learned cardinalities: Estimating correlated joins with deep learning. In CIDR, 2019.
8. B.Ding,S.Das,R.Marcus,W.Wu,S.Chaudhuri,and.R.Narasayya.AI meet sAI: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 concurrent queries using graph embedding. VLDB, 13(9):1416–1428, 2020.
10. C. Zhang, R. Marcus, A. Kleiman, and O. Papaemmanouil. Buffer pool aware query 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 Wolfgang Lehner. 2021. PostCENN: postgreSQL with machine learning models for cardinality estimation. Proc. VLDB Endow. 14, 12 (July 2021), 2715–2718.
13. Ryan Marcus, Parimarjan Negi, Hongzi Mao, Nesime Tatbul, Mohammad Alizadeh, and Tim Kraska. 2021. Bao: Making Learned Query Optimization Practical. In Proceedings 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 Neoklis Polyzotis. 2018. The Case for Learned Index Structures. In Proceedings of the 2018 International Conference on Management of Data (SIGMOD '18). Association for Computing Machinery, 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 Proceedings of the 9th Biennial Conference on Innovative Data Systems Research (CIDR 2019). MIT DSpace.
16. Xiaonan Nie, Xupeng Miao, Zilong Wang, Zichao Yang, Jilong Xue, Lingxiao Ma, Gang Cao, and Bin Cui. 2023. FlexMoE: Scaling Large-scale Sparse Pre-trained Model Training via Dynamic Device Placement. Proc. ACM Manag. Data 1, 1, Article 110 (May 2023), 19 pages.

Thank you.

Presented By:

Darshit Amit Pandya

Find the Slides at: darshit-pandya.github.io/AIC2024.pdf

Divert any Questions to: darshitpandya211@gmail.com