تحسين استعلامات SQL باستخدام التعلم العميق
Optimizing SQL Queries Using Deep Learning

١. الملخص:

تستعرض المقالة تطور تحسين أداء استعلامات SQL من المحسنات التقليدية إلى أنظمة ذكية تعتمد على التعلم العميق، حيث عانت المحسنات التقليدية من أخطاء في تقدير الكاردينياليتي(Cardinality Estimation) والتي تمثل عدد الصفوف الناتجة عن عمليات الربط والتصفية وذلك خاصة في البيانات المعقدة، لذلك ظهرت نماذج التعلم العميق مثل NeuroCard .

كما قدمت النماذج الهجينة التي دمجت بين المحسنات التقليدية القائمة على التكلفة ونماذج التعلم الآلي أو نماذج اللغات الكبيرة (LLMs) واستغلت نقاط القوة في كل نهج، أنظمة مثل Bao (Bandit-based Adaptive Query Optimizer) تحقق تسريعاً يصل إلى ٨ أضعاف في الاستعلامات المعقدة دون تعديل المحسن التقليدي، وفي نفس المجال استخدم نظام CodexDB نماذج لغوية (مثل GPT-3) لترجمة الاستعلامات إلى شيفرة برمجية.

بالنتيجة أثبتت نماذج التعلم العميق جدواها في البيئات السحابية مثل BigQuery, Azure SQL، بيد أن ثمة تحديات قائمة، من أبرزها الاعتماد على واجهات خارجية وارتفاع في تكلفة التدريب، لذلك يتوقع اتجاه الأبحاث الحديثة نحو أنظمة هجينة قابلة للتفسير وذاتية الإدارة.

٢. مقدمة:

في ظل التنامي المتسارع لحجوم البيانات وتعقيدها، يبرز تحسين أداء استعلامات لغة الاستعلام البنيوية (SQL) كأحد الركائز الأساسية لضمان كفاءة أنظمة قواعد البيانات وسرعة استجابتها، حيث لم يعد هذا التحسين مقتصراً على كونه ممارسة تقنية تكميلية، بل أصبح عاملاً حاسماً يؤثر مباشرة على الأداء التشغيلي للمنظمات، وصولاً إلى تحسين مؤشراتها المالية وسمعتها التنافسية. تُظهر الدراسات أن انخفاض أداء الأنظمة أو تعطلها ولو لفترات وجيزة يمكن أن يتسبب في خسائر فادحة للشركات، خاصة في قطاعات حساسة للوقت مثل التجارة الإلكترونية والخدمات المالية، حيث قد تصل التكلفة إلى آلاف الدولارات في الدقيقة الواحدة، وبالتالي يتحول تحسين استعلامات SQL إلى استثمار استراتيجي يدعم مرونة الأعمال واستدامتها، فهو يمكّن المؤسسات من استخلاص الرؤى من بياناتها في الوقت المناسب، ويدعم اتخاذ القرارات السريعة والدقيقة، مما يعزز موقعها التنافسي في سوق قائم على التحول الرقمي والاعتماد المكثف على البيانات. لذلك سوف نستعرض المحسنات لأداء الاستعلامات، نبدأ من المحسنات التقليدية ومن ثم المحسنات المعتمدة على التعلم العميق.

٢.١ أهداف المقالة:

تهدف هذه المقالة إلى مراجعة تطور أداء محسنات استعلامات SQL من خلال:

- تقييم المحسنات التقليدية القائمة على القواعد (RBO) والتكلفة (CBO) وتحديد نقاط ضعفها الأساسية كمشكلة تقدير الكاردينياليتي(Cardinality Estimation) وتراكم أخطائها في بيئات البيانات المعقدة.
- تحليل النماذج الحديثة المعتمدة على التعلم العميق والتعلم المعزز مثل (NeuroCard, Fauce, Bao) وفهم البنية المعمارية البرمجية لهذه النماذج.
- استعراض النماذج الهجينة المستفيدة من نماذج اللغات الكبيرة (LLMs) مثل نظام (CodexDB) القائم على GPT-3 في معالجة الاستعلامات وتحويلها إلى شيفرات برمجية.
- مقارنة محورية بين النماذج المدروسة من حيث كفاءة الأداء والقيود التشغيلية.

٣.المحسنات التقليدية لأداء استعلامات SQL:

٣.١ التحسين القائم على القواعد ((Rule-Based Optimization (RBO)): [1]

يستند هذا النهج إلى تطبيق مجموعة محددة مسبقاً من قواعد التحويل الجبرية العلائقية للخطة المنطقية للاستعلام، مع قواعد محددة للخطة الفيزيائية، أي يعمل محسن الاستعلام وفق استراتيجية ثابتة للتطبيق دون الاعتماد على تقدير تكلفة تنفيذية فعلية. يُفضل هذا التحسين في بيئات البيانات الموزعة والضخمة ذات الطبيعة الديناميكية. مثل PrestoDB و SparkSQL التي تعتمد على نُهج تحسين يبدأ بقواعد أساسية قبل أي تقدير للتكلفة، كما يستخدم في الاستعلامات الأولية في أنظمة مثل Apache Calcite. يمتاز هذا المحسن بالسرعة في توليد خطط التنفيذ بالإضافة إلى سهولة الصيانة، نظراً لطابعه الحتمي وعدم اعتماده على نماذج تكلفة معقدة أو إحصائيات ديناميكية، إلا أنه غالباً ما ينتج خطط تنفيذ دون المستوى الأمثل في البيانات غير المنتظمة أو متغيرة الحجم.

٣.٢ التحسين القائم على التكلفة ((Cost-Based Optimization (CBO)): [2]

يمثل هذا النهج النموذج الأكثر تطوراً في أنظمة قواعد البيانات العلائقية التقليدية، حيث يولد المحسن فضاء بحثياً واسعاً من الخطط البديلة، ويقدر تكلفة كل منها باستخدام نموذج تحليلي لمحاكاة استخدام الموارد (المعالج، الذاكرة، القرص، الشبكة). يعتمد اختيار التنفيذ الأمثل على دقة تقدير الكارديناليتي(Cardinality Estimation) (عدد الصفوف الناتجة عن عمليات الربط والتصفية) بينما تبقى أخطاء هذا التقدير في عمليتي الوصل والتصفية هي العامل الرئيسي في ضعف الاداء. يستخدم هذا المحسن في محسنات أنظمة مثل PostgreSQL و Microsoft SQL Server و Oracle وفي البيئات التحليلية الموزعة، و يُعد محسن Orca المستخدم في Greenplum و HAWQ مثالاً متقدماً على CBO مصمم خصيصاً للبيئات الموزعة .يوفر هذا المحسن إحصائيات دقيقة تمكنه من توليد خطط تنفيذ قريبة جداً من الأمثلية، مع أخذ خصائص البيانات والبنية التحتية في الاعتبار ، إلا أنه يعاني من مشكلة التبعية (Error Propagation)، حيث تتفاقم أخطاء تقدير الكارديناليتي الأولية عبر مراحل تنفيذ الخطة، مما قد يؤدي إلى اختيار خطط كارثية ، كما أن تعقيد عملية البحث والتقدير يجعله أكثر استهلاكاً للوقت والذاكرة مقارنة بـ RBO.

٤. التعلم العميق في تحسين أداء استعلامات SQL: [4][3]

سنعرض اولاً تمثيل الاستعلامات رقمياً، حيث تعتبر هذه الخطوة الأساسية لتطبيق التعلم العميق في تحسين أداء الاستعلامات.

٤.١ تمثيل الاستعلامات رقمياً:

لا تفهم الشبكات العصبية النص المكتوب بلغة SQL مما يستدعي تمثيلها بمتجه رقمي ثابت الأبعاد يحافظ على دلالات الاستعلام وبنيته المنطقية، وذلك بطرق عديدة منها:

a. تمثيل النص الصّرف (SQL Text): يستخدم لذلك نماذج لغوية مُدرّبة على الشيفرة البرمجية لتضمين نص الاستعلام. تمتاز هذه الطريقة بالسرعة إلا أنها تتجاهل البنية المنطقية للاستعلام.
b. تمثيل خطة التنفيذ: تعتبر الطريقة الأهم لأن المحسنات تعتمد على خطة التنفيذ التي تكون على شكل شجرة من أبرزها ثلاث معماريات لترميز هذه الشجرة: LSTM-Tree، CNN-Tree، Graph Neural Networks(GNNS)
c. تمثيل بنية الاستعلام: من أمثلتها نموذج MSCN الذي يمثل الاستعلام كثلاث مجموعات مستقلة هي الجداول وشروط الربط والفلاتر ويُستخلص من كل مجموعة متجه ثم تُدمج في متجه واحد.
d. تمثيلات غنية بالسياق: تقوم هذه الطريقة بتمثيل المخطط (مفاتيح خارجية، أنواع أعمدة،...) باستخدام عدة نماذج لغوية، من أمثلتها GTDD (حيث استخدم شبكات بيانية لمخطط الاستعلام و Tree-LSTM لخطة التنفيذ).

٤.٢ نماذج المحسنات المعتمدة على التعلم العميق:

يمثل استخدام التعلم العميق في مجال تحسين أداء الاستعلامات توجهاً حديثاً يهدف إلى تجاوز مشكلات المحسنات التقليدية القائمة على التكلفة والناتجة عن عدم قدرتها على نمذجة الارتباطات بين بيانات الجداول المختلفة ومن أبرزها:

- **نماذج تقدير الكارديناليتي المُتعلَّمة (Learned Cardinality Estimation Models)**:[3]

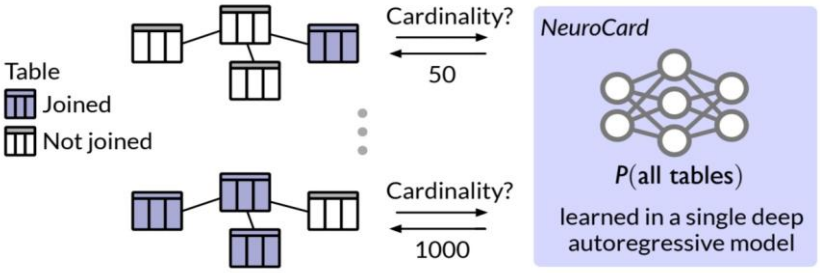
ظهرت هذه الفئة من النماذج كاستجابة مباشرة لأهمية ودقة تقدير الكارديناليتي في أنظمة CBO، حيث تستبدل النماذج الإحصائية التقليدية بنماذج تعلم آلي مدربة على عينات من البيانات منها نماذج (Sum–Product Networks) كما في DeepDB، ونماذج مثل Neurocard MSCN، تهدف هذه النماذج إلى التقاط الارتباطات المعقدة بين السمات التي تتجاهلها الإحصائيات التقليدية لتحسين دقة التقدير. أظهر هذا النوع من النماذج قدرة على تحقيق دقة تقدير أعلى مقارنةً مع الطرق التقليدية للبيانات المعقدة مما يؤدي إلى تحسين جودة خطط التنفيذ النهائية، من أشهر هذه النماذج:

- **نموذج (Neural Estimation Unified Regression Optimization) NeuroCard**: [5]

يمثل هذا النموذج تحولاً من النماذج المعتمدة على الفرضيات التقليدية إلى النماذج المعتمدة على التعلم العميق حيث يوفر لمحركات قواعد البيانات وسيلة دقيقة لاختيار أفضل خطط تنفيذ للاستعلامات في البيئات الضخمة والمعقدة.

مبدأ عمله:

يعمل نموذج NeuroCard على إنشاء مِقدّر كثافة عصبي موحد (Unified Neural Density Estimator) صُمم خصيصاً لاستيعاب التوزيع الاحتمالي المشترك لجميع سجلات جداول قاعدة البيانات من خلال كيان إحصائي واحد. ويتم هذا التوزيع من خلال تدريب النموذج على عينات ماخوذة من الربط الخارجي لجميع الجداول، مما يمنحه قدرة فائقة على رصد أدق الارتباطات والاعتمادات المتبادلة بين أعمدة الجداول المختلفة دون الحاجة لفرضيات استقلال مسبقة. وبمجرّد اكتمال مرحلة التدريب، يصبح النموذج قادراً على تقدير الكارديناليتي لأي استعلام وذلك من خلال إجراء استدلال إحصائي فوري على بنية النموذج الواحد، ويوضح الشكل –١– ألية عمل نموذج NeuroCard



الشكل–١– يوضح ألية عمل نموذج NeuroCard في تقدير الكارديناليتي(Cardinality) من خلال نموذج انحدار ذاتي عميق موحد(single deep autoregressive model) على بيانات الجداول المترابطة (Table Joined) وغير المترابطة (Table Not Joined)

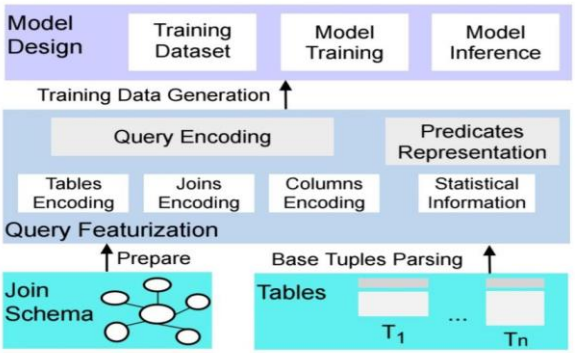
- **نموذج (Fast and Accurate Uncertainty–aware Cardinality Estimation) Fauce**: [6]

يعد نموذج Fauce قفزة نوعية في مجال تقدير الكارديناليتي داخل قواعد البيانات من خلال قدرته على استيعاب الارتباطات بين كافة الأعمدة والجداول، وكونه الأول من نوعه الذي يدمج معلومات عدم اليقين (uncertainty) ضمن تقديراته مما يعزز من موثوقية النتائج.

مبدأ عمله:

يقوم بتدريب عدة شبكات عصبية مستقلة لكن بأوزان ابتدائية مختلفة حيث تتنبأ كل شبكة بقيمة، ومن ثم يقاس عدم اليقين للنموذج من خلال تباين قيم التنبؤات وبناء على مستويات عدم اليقين إما أن يعتمد التقدير أو يزاد التدريب أو توسع بيانات التدريب، وهذا يساعد على تقليل أخطاء تقدير الكارديناليتي ، وخاصة في الاستعلامات المعقدة التي تحوي على ترابطات متعددة ، لكن بالرغم أن Fauce مكون من عدة شبكات إلا أنه صمم ليكون خفيفاً في الحجم مما يجعله أسرع بعشرة مرات من النماذج الأخرى، بالإضافة لكونه مؤلف من تجميع عدة شبكات يساعده في التعلم على

العلاقات المتداخلة بين الجداول و الأعمدة بشكل أكثر عمقاً من الشبكات الفردية. يوضح الشكل -٢- ألية عمل نموذج Fauce



الشكل-٢- يوضح ألية عمل نموذج Fauce المعتمد على الجداول (Tables) ومخطط الربط (Join Schema) فيما بينها، حيث تستخلص الخصائص المميزة للاستعلام (Query Featurization) من خلال ترميز الاستعلام (Query Encoding) الذي يشمل ترميز الجداول (Tables Encodig) وترميز علاقات الربط (Joins Encoding) وترميز الأعمدة (Columns Encoding) ، بالإضافة إلى استخراج المعلومات الإحصائية (Statistical Information)، وتمثل هذه مرحلة توليد بيانات التدريب (Training Data Generation)، بينما مرحلة تصميم النموذج (Model Design) تعتمد على مجموعة بيانات التدريب (Training Dataset) ثم تدريب النموذج (Model Training) وصولاً إلى توقع النموذج (Model Inference) .

٥. التعلم المعزز العميق:

تظهر الأبحاث الحديثة أن نماذج التعلم العميق يمكن أن تحقق تحسينات كبيرة في أداء الاستعلامات مقارنة بالمحسنات التقليدية، إلا أنه لم يتم دمج معظم هذه النماذج في قواعد البيانات الإنتاجية، ويرجع ذلك إلى التحديات العملية المرتبطة ببناء وتكامل وصيانة هذه الأنظمة المعقدة، كما أن المحسنات التقليدية تستهلك زمناً طويلاً لاختيار خطة التنفيذ المثلى للاستعلام ، من هنا ظهرت الحاجة إلى نماذج التعلم المعزز العميق التي يتعلم فيها الوكيل من خلال التجربة و الخطأ كيفية اختيار الخطط التي تقلل زمن التنفيذ ، ومن أبرزها:

- **نظام Bao (Bandit-based Adaptive Query Optimizer)**: [2]

يمثل Bao نقلة نوعية نحو التحسين التكيفي الذاتي للاستعلامات، حيث تتعلم الأنظمة من تجربتها السابقة لتحسين أدائها المستقبلي بشكل مستمر، ويُعد Bao أول محسن استعلامات متعلم يتم دمجه بنجاح في نظام قاعدة بيانات إنتاجي مفتوح المصدر (PostgreSQL). يتميز Bao باتباع نهج هجين عملي يتمثل بالحفاظ على المحسن التقليدي حيث يترك Bao المحسن التقليدي القائم على التكلفة (CBO) في PostgreSQL سليماً ويعمل بشكل طبيعي، ثم يستخدم التعلم الآلي فقط للتنبؤ بأداء خطط التنفيذ المختلفة التي يولدها المحسن التقليدي تحت مجموعات مختلفة من التلميحات أو إعدادات التحسين.

يمتاز Bao بالتكيف السريع حيث يمكن أن يتعلم اختيار تلميحات عالية الجودة لاستعلامات غير مرئية سابقًا في زمن قياسي، كما أظهر تفوقاً على المحسنات التقليدية والمتعلمة الأخرى في مهام معيارية صعبة منها في قاعدة بيانات IMDb وصلت إلى ٨ أضعاف في تحسينات الأداء لبعض الاستعلامات المعقدة.

لكن من قيود نظام Bao أنه لا يحل مشاكل المحسن الأساسي أي لا يصحح أخطاء تقدير الكارديناليتي الأساسية في المحسن التقليدي، بل فقط يوجهه إلى اختيار خطط بديلة تقلل من الأخطاء الحاصلة في المحسنات التقليدية.

٦. النماذج الهجينة: [7]

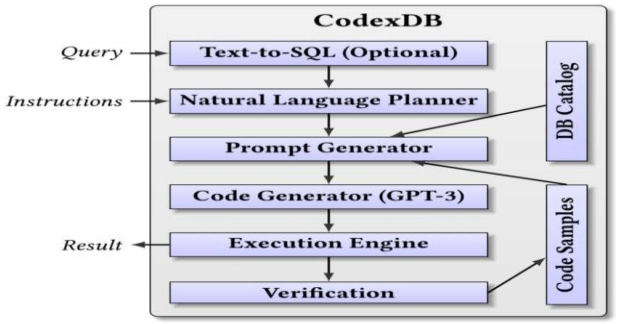
تمثل النماذج الهجينة اتجاهاً حديثاً في مجال تحسين استعلامات قواعد البيانات حيث تسعى إلى تجاوز المشاكل من خلال دمج المحسنات التقليدية القائمة على التكلفة وتقنيات التعلم الآلي أو نماذج اللغات الكبيرة (LLMs) واستغلال نقاط القوة في كل نهج، في هذا المجال يبرز نظام CodexDB كأحد أبرز الأمثلة على دمج بين نماذج اللغات الكبيرة وتقنيات معالجة قواعد البيانات التقليدية.

٦.١ نظام CodexDB: [9] [8]

يعتمد CodexDB على بنية يقوم فيها مخطط استعلامات SQL التقليدي بتحليل الاستعلامات المعقدة إلى سلسلة من خطوات المعالجة البسيطة الموصوفة باللغة الطبيعية (مثل "اقرأ الجدول X"، "صفّ النتائج حسب الشرط Y")، ثم دمج هذه الخطوات بعد ذلك بتعليمات المستخدم النصية لتشكل موجهاً إلى نموج توليدي GPT-3 Codex الذي يقوم بترجمة النص الناتج إلى شفرة برمجية (مثل كود Python مثلاً) قابلة للتنفيذ.

أظهر النموذج الأولي قدرته على توليد شيفرة برمجية صحيحة بنسبة ٧١٪ لاستعلامات من معيار WikiSQL. ومع ذلك نظهر تحديات يعاني منها نظام codexDB تكمن بالاعتماد على نموذج خارجي GPT-3 Codex تتعلق بالتكلفة والخصوصية، بالإضافة إلى اقتصر التقييم الأولي على استعلامات أحادية الجدول (WikiSQL) حيث لم يغطِ استعلامات الربط (JOIN) المعقدة.

يوضح الشكل -٣- رسم تخطيطي مبسط لمعمارية CodexDB تظهر تدفق الاستعلام من SQL إلى خطوات طبيعية ثم إلى شيفرة برمجية عبر نموذج لغة LLMs.



الشكل -٣- يوضح معمارية CodexDB المؤلفة من خطوة الاستعلام (Query) التي تحويل النص إلى استعلام (Text-to-SQL)، ثم خطوة التعليمات التوجيهية (Instructions) التي تتم من مرور الاستعلام على مخطط اللغة الطبيعية (Natural Language Planner)، ثم مولد الأوامر (Prompt Generator) والذي يعتمد على مصدرين أساسيين هما دليل قاعدة البيانات (DB Catalog) وعينات الشيفرات البرمجية (Code Samples)، وأخيراً مولد الشيفرة البرمجية (Code Generator) المعتمد على نموذج GPT-3، أما خطوة النتيجة (Result) تعتمد على محرك التنفيذ (Execution Engine) و وحدة التحقيق (Verification).

٧. مقارنة تحليلية لنماذج تحسين أداء استعلامات SQL:

سنقدم مقارنة تحليلية بين نماذج تحسين أداء الاستعلامات المذكورة سابقاً وذلك من خلال معايير جوهرية هي: المدخلات، البنية العصبية، نقاط القوة، القيود الحالية، وأبرز أماكن التطبيق العملي الذي يحدد مدى قابلية كل نموذج للدمج في أنظمة حقيقية

اسم النموذج	المدخلات	المنهجية المعتمدة	نقاط القوة	القيود الحالية	التطبيق العملي
NeuroCard	كامل عينات الجداول بعد الربط الخارجي الكامل .	نموذج انحدار ذاتي موحد يدمج جميع الجداول في كيان احصائي واحد.	تحسن كبير ودقة عالية في تقدير الارتباطات المعقدة بين الأعمدة من جداول مختلفة.	تكلفة تدريب عالية. صعوبة التحديث الديناميكي عند تغير البيانات بشكل متكرر.	مجالات بحثية على PostgreSQL، ولم يدمج تجارياً.
Fauce	عينات من الأعمدة، والجداول المتداخلة.	شبكات عصبية متعددة ومستقلة.	تقليص أخطاء التقدير، ودمج معايير عدم اليقين.	زيادة تكلفة التدريب الأولي بسبب تعدد الشبكات، بالرغم من سرعة الاستدلال العالية.	بيئات تجريبية، ولم يطرح في أنظمة مفتوحة المصدر.
Bao	الاستعلامات المنفذة مسبقاً مع زمن تنفيذها الفعلي.	شبكة عصبية شجرية تلافيفية مقترنة بخوارزمية التعلم المعزز العميق.	تسريع يصل إلى 8 أضعاف في بعض الاستعلامات المعقدة.	يعتمد على توجيهات المحسن التقليدي ولا يحل مشاكله الأساسية المتعلقة بتقدير الكاردينالتي.	مدمج فعلياً في PostgreSQL كنظام مفتوح المصدر.
CodexDB	استعلامات اللغة الطبيعية ومخطط قاعدة البيانات بالإضافة إلى عينات من الشيفرات البرمجية	نموذج لغوي كبير (LLM)، وتحديداً GPT-3 / Codex.	دقة تصل إلى 71% في توليد شيفرات برمجية صحيحة.	تكلفة مادية مرتفعة لاعتماده على (API)خارجي ، و زمن استجابة مرتفع، وعدم دعم استعلامات (JOIN) المعقدة	نماذج أولية، ولكن غير مؤهل للإنتاج الفعلي بسبب التكلفة وزمن التأخير

الجدول -١- يوضح مقارنة لنماذج تحسين أداء استعلامات SQL

يتضح من المقارنة أن تطور محسنات استعلامات SQL لم يكن استبدال مباشر للمحسنات التقليدية، بل جاء لمعالجة نقاط ضعف محددة من أهمها مشكلة تقدير الكاردينياليتي. فقد حقق NeuroCard أعلى دقة قي تقدير الارتباطات المعقدة بين الجداول إلا أن اعتماده على تدريب مكلف وصعوبة تحديثه مع تغير البيانات يجعلان استخدامه أكثر ملائمة للبيئات البحثية، بينما يركز Fauce على زيادة موثوقية التقدير من خلال دمج مفهوم عدم اليقين مما يجعله أكثر استقراراً، وإن كان ذلك على حساب زيادة زمن المعالجة نتيجة استخدام عدة نماذج عصبية.

أما Bao لا يسعى إلى تحسين الكاردينياليتي وإنما يعمل فوق المحسن التقليدي لاختيار أفضل خطة تنفيذ بالاعتماد على التعلم المعزز وهذا يمنحه ميزة سهولة الدمج داخل أنظمة قواعد البيانات القائمة.

بينما يمثل CodexDB اتجاهاً مختلفاً حيث يعتمد على نماذج اللغات الكبيرة وينقل عملية التحسين من اختيار خطة تنفيذ الاستعلام إلى توليد الشيفرات البرمجية له، لكن بالرغم من الإمكانيات الكبيرة لهذا النهج إلا أن اعتماده على نماذج خارجية يفرض تحديات تتعلق بالتكلفة وزمن الاستجابة ويجعله غير مؤهل للإنتاج الفعلي.

مما سبق نستنتج أن اختيار النموذج الأمثل يعتمد على طبيعة بيئة العمل، ففي البيئات البحثية التي تتطلب أعلى دقة في تقدير الكاردينياليتي فإن NeuroCard الخيار الأنسب، بينما يعد Bao أكثر ملائمة للأنظمة الإنتاجية التي تحتاج إلى تحسين الأداء دون تعديل جذري لمحرك قاعدة البيانات، أما CodexDB فيمثل توجهاً مختلفاً ولا يزال بحاجة إلى تطوير لتجاوز القيود التشغيلية الحالية.

٨. التطبيقات العملية في الأنظمة الحديثة:[10][2][1]

٨.١ أنظمة قواعد البيانات السحابية:

تشهد خدمات قواعد البيانات السحابية القابلة للتوسع اندماجاً عميقاً لتقنيات التعلم الآلي، مما يحولها من منصات تخزين ومعالجة إلى أنظمة ذاتية التعلم والتكيف، منها:

- **Google BigQuery :**

يعتمد Google BigQueryعلى تقنيات التعلم الآلي في عدة مكونات أساسية لمعالجة الاستعلامات، حيث تستخدم بنيته الموزعة نماذج تعلم آلي لتحسين دقة تقدير الكارديناليتي التي تعد من أبرز مصادر الخطأ في خطط التنفيذ، كما تستخدم نماذج التعلم الآلي لتحسين خطط التنفيذ وتقليل الهدر في الموارد الحسابية. يعد هذا النهج نقلة نوعية في تطوير محركات قواعد البيانات، حيث يمكن النظام من التعلم المستمر وتحقيق توازن أفضل بين الأداء العالي والتكلفة التشغيلية.

- **Amazon Aurora :**

ركزت شركة Amazon على دمج الذكاء الاصطناعي في خدمات قواعد البيانات السحابية الخاصة بها، خاصة في نظام Amazon Aurora حيث تعتمد مُحسّن الاستعلام الذكي لتحليل أنماط الاستعلامات والتنبؤ بكمية الموارد التي سيحتاجها الاستعلام والذي يؤدي إلى تنفيذ أكثر كفاءة، كما تستفيد Aurora من أدوات مثل PI Reporter لإعداد تقارير مفصلة عن أداء قواعد البيانات و تحديد نقاط التحسين على PostgreSQL .

٨.٢ أنظمة مفتوحة المصدر:

يتم دمج تقنيات الذكاء الاصطناعي في الأنظمة مفتوحة المصدر عبر الإضافات أو التعديلات على الأطر القائمة، مما يجعل هذه التقنيات متاحة على نطاق أوسع، منها:

- **إضافات الذكاء الاصطناعي لـ PostgreSQL:**

انتقلت PostgreSQL من كونها قاعدة بيانات علائقية تقليدية إلى منصة هجينة تدعم الذكاء الاصطناعي، حيث تعد إضافة pgvector الأكثر أهمية للمساعدة في بناء قواعد بيانات المتجهات (Vector Databases)، كما تسمح بتخزين والبحث في التمثيلات المتجهية الناتجة عن نماذج اللغات الكبيرة وتطبيق البحث الدلالي .

بالإضافة إلى ذلك، ظهرت إضافات أخرى متقدمة مثل NeuroCard لتحسين تقدير الكارديناليتي, وإضافات تتيح دمج نماذج التعلم الآلي مباشرة داخل دوال .SQL .

- **مُحسِن Apache Calcite:**

يعتبر Apache Calcite العمود الفقري للعديد من أطر معالجة البيانات المفتوحة المصدر مثل Hive و Flink ، حيث يركز البحث الحالي على دمج التعلم الآلي فيه لتحسين المخططات التنفيذية، ومن أبرز الأعمال في هذا المجال نموذج QueryFormer وهو نموذج مستوحى من بنية المحولات(Transformer) متخصص في تعلم تحسين الاستعلامات.

٩. الخاتمة:

أظهرت هذه الدراسة أن تحسين أداء استعلامات SQL قد شهد تحولاً من النماذج التقليدية إلى أنظمة ذكية قائمة على التعلم العميق، حيث أثبتت المحسنات التقليدية فعالية في بيئات محددة لكنها عانت من قيود مثل أخطاء تقدير الكارديناليتي المتراكمة وعدم قدرتها على نمذجة الارتباطات المعقدة بين البيانات، بينما أظهرت النماذج المُتعلمة مثل NeuroCard دقة عالية في تقدير الكارديناليتي، كما قدم نظام Bao نهجاً هجيناً متميزاً يحافظ على المحسن التقليدي ويُحسّن أداءه عبر التعلم التكيفي

محققاً تسريعات تصل إلى ٨ أضعاف في استعلامات معقدة. اما نماذج اللغات الكبيرة مثل CodexDB فقد فتحت آفاقاً جديدة لترجمة الاستعلامات إلى شيفرة برمجية، رغم تحديات الخصوصية والتكلفة.

على الصعيد التطبيقي، نجحت أنظمة سحابية مثل BigQuery و Azure SQL في دمج هذه التقنيات لتحقيق جدولة ذكية وتوزيع متوازن لأعباء العمل، مما يؤكد جدوى الحلول المتعلمة في بيئات الإنتاج.

ورغم هذه التطورات النوعية، لا تزال هناك تحديات معقدة تعيق التطبيق التجاري الواسع لهذه النماذج تكمن في تكلفة التشغيل حيث النماذج المعتمدة على واجهات برمجة التطبيقات الخارجية كما في CodexDB مع GPT-3 تتطلب عبئاً مالياً متصاعداً مع زيادة عدد الاستعلامات، وذلك مقارنة بالمحسنات التقليدية المجانية تماماً والمدمجة محلياً داخل محركات قواعد البيانات، كما أن التأخير في زمن الاستجابة الناتج عن تأخيرات النماذج السحابية يمنع من تطبيق هذه الحلول في بيئات الإنتاج اللحظية ويجعل المحسنات التقليدية خياراً أكثر استدامة حتى الآن.

لذا يُتوقع أن تتجه الأبحاث المستقبلية على تطوير محسنات هجينة أكثر كفاءة تجمع بين المحسنات التقليدية وتقنيات الذكاء الاصطناعي، مع تحسين قابلية التفسير، وتقليل تكلفة التدريب ودعم التعلم المستمر للتكييف مع تغير البيانات. ومن شأن هذه التوجهات أن تسهم في بناء أنظمة أكثر ذكاءً وقابلة للتطبيق في البيئات الإنتاجية.

١٠. المراجع:

[1] Leis, V., Gubichev, A., Mirchev, A., Boncz, P., Kemper, A., & Neumann, T. "How Good Are Query Optimizers, Really?", VLDB Endowment, Vol. 18, No. 12 ISSN 2150-8097,(2015).

[2] Marcus, R., Negi, P., Mao, H., Tatbul, N., Alizadeh, M., & Kraska, T. "Bao: Making Learned Query Optimization Practical." Proceedings of the 2021 International Conference on Management of Data (SIGMOD), (2021).

[3] Han, Y., Wu, Z., & Wu, P., et al. "Cardinality Estimation in DBMS: A Comprehensive Benchmark Evaluation", Proceedings of the VLDB Endowment, Vol. 15, No. 4 ISSN 2150-8097. (2021).

[4] Marcus, R., & Papaemmanouil, O. "Deep Reinforcement Learning for Join Order Enumeration." Proceedings of the 1st International Workshop on Exploiting Artificial Intelligence Techniques for Data Management, ACM ISBN 978-1-4503-5851-4, (2018).

[5] Yang, Z., et al. "NeuroCard: One Cardinality Estimator for All Tables." , Proceedings of the VLDB Endowment, Vol. 14, No. 1, (2020).

[6] Wu, P., et al. "Fauce: Fast and Accurate Uncertainty-aware Cardinality Estimation." Proceedings of the VLDB Endowment,vol. 15,No.11,(2022) .

[7] Yu, X., et al. "Cost-based or Learning-based? A Hybrid Query Optimizer for Query Plan Selection", Proceedings of the VLDB Endowment, Vol. 15, No. 13 ISSN 2150-8097, (2023).

[8] Trummer, I. "CodexDB: Synthesizing Code for Query Processing from Natural Language Instructions using GPT-3 Codex", Proceedings of the VLDB Endowment, Vol. 15, No. 11 ISSN 2150-8097, (2022).

[9] Chen, M., et al. "Evaluating Large Language Models Trained on Code." arXiv preprint arXiv:2107.03374v2, (2021).

[10] Woltmann, L., Olwig, D., Hartmann, C., Habich, D., Lehner, W.," PostCENN: PostgreSQL with Machine Learning Models for Cardinality Estimation ", Proceedings of the VLDB Endowment, Vol. 14, No. 12 ISSN 2150-8097,(2021).