Reading & Presenting Papers

CIS 6500

Ryan Marcus

Outline

- Seminar specifics
- How to read a DB paper
  - The 3 pass method
  - Murat’s method
- How to give a good seminar talk

Outline

- **Seminar specifics**
- How to read a DB paper
  - The 3 pass method
  - Murat’s method
- How to give a good seminar talk

Paper Assignments

- https://rm.cab/papers
- Please respond by Monday, 9/15 10am

Long Report

- 1 week* before your seminar, “long report” due
  - 2-3 pages, should contain a good example.
- Proves to me you understood the paper
- Distributed at the end of the semester

* 3 days before for first two seminars

Short Report

- Before noon on each seminar day, “short report” due
- 300-500 words

Your Seminar Structure

Time	Content
noon – 12:10pm	Prep
12:10pm – 12:15pm	Announcements (Ryan)
12:15pm – 1:00pm	Lecture (you)
1:00pm – 1:30pm	Discussion (you)

Outline

- Seminar specifics
- **How to read a DB paper**
  - The 3 pass method
  - Murat’s method
- How to give a good seminar talk

How DB Papers Work

- Authors have *page limits*
  - Style becomes condensed
- Papers are *peer reviewed*
  - Audience is other researchers in the field

How DB Papers Work

- Generally four types of papers
- Conference papers (90%)
  - Strict page limits (12 pages)
  - Most visible papers
  - Important: SIGMOD, VLDB, CIDR, ICDE
- Journal papers (5%)
  - Generally extensions of conference papers
  - No (or very generous) page limits
  - Important: JACM, VLDBJ, TODS

How DB Papers Work

- Workshop papers (2.5%)
  - 4-8 pages
  - Peer reviewed, but early work. Presenting an idea.
- Technical reports (2.5%)
  - Internal documents, no peer review
  - No page limits
  - Good source of details

Reading your paper: Title

Optimizing Data-intensive Systems in Disaggregated Data Centers with TELEPORT

Qizhen Zhang, Xinyi Chen, Sidharth Sankhe, Zhilei Zheng, Ke Zhong, Sebastian Angel, Ang Chen[§], Vincent Liu, Boon Thau Loo
University of Pennsylvania, [§]Rice University
{qizhen, cxinyic, sankhe, zhileiz, kezhong, sga001, liuv, boonloo}@seas.upenn.edu, [§]angchen@rice.edu

Reading your paper: Title

Optimizing Data-intensive Systems in Disaggregated Data Centers with TELEPORT

Qizhen Zhang, Xinyi Chen, Sidharth Sankhe, Zhilei Zheng, Ke Zhong, Sebastian Angel, Ang Chen[§], Vincent Liu, Boon Thau Loo
University of Pennsylvania, [§]Rice University
{qizhen, cxinyic, sankhe, zhileiz, kezhong, sga001, liuv, boonloo}@seas.upenn.edu, [§]angchen@rice.edu

- First author, main contributor, “owner” of the project.
- Normally a student or postdoc of the last author

Reading your paper: Title

Optimizing Data-intensive Systems in Disaggregated Data Centers with TELEPORT

Qizhen Zhang, Xinyi Chen, Sidharth Sankhe, Zhilei Zheng, Ke Zhong, Sebastian Angel, Ang Chen[§], Vincent Liu, Boon Thau Loo
University of Pennsylvania, [§]Rice University
{qizhen, cxinyic, sankhe, zhileiz, kezhong, sga001, liuv, boonloo}@seas.upenn.edu, [§]angchen@rice.edu

- Other “junior” authors (Ph.D., undergrad, postdoc)
- Contribution order

Reading your paper: Title

Optimizing Data-intensive Systems in Disaggregated Data Centers with TELEPORT

Qizhen Zhang, Xinyi Chen, Sidharth Sankhe, Zhilei Zheng, Ke Zhong, Sebastian Angel, Ang Chen[§],
Vincent Liu, Boon Thau Loo
University of Pennsylvania, [§]Rice University
{qizhen, xinyic, sankhe, zhileiz, kezhong, sga001, liuv, boonloo}@seas.upenn.edu, [§]angchen@rice.edu

- Secondary senior authors

Reading your paper: Title

Optimizing Data-intensive Systems in Disaggregated Data Centers with TELEPORT

Qizhen Zhang, Xinyi Chen, Sidharth Sankhe, Zhilei Zheng, Ke Zhong, Sebastian Angel, Ang Chen[§], Vincent Liu, Boon Thau Loo
University of Pennsylvania, [§]Rice University
{qizhen, xinyic, sankhe, zhileiz, kezhong, sga001, liuv, boonloo}@seas.upenn.edu, [§]angchen@rice.edu

- Primary senior author (generally advisor of 1st author)
- $$$

Reading your paper: Title

Check Out the Big Brain on BRAD: Simplifying Cloud Data Processing with Learned Automated Data Meshes

Tim Kraska*
MIT CSAIL
Amazon Web Services
kraska@mit.edu
timkrask@amazon.com

Tianyu Li*
MIT CSAIL
litianyu@mit.edu

Samuel Madden*
MIT CSAIL
madden@csail.mit.edu

Markos Markakis*
MIT CSAIL
markakis@mit.edu

Amadou Ngom*
MIT CSAIL
ngom@mit.edu

Ziniu Wu*
MIT CSAIL
ziniuw@mit.edu

Geoffrey X. Yu*
MIT CSAIL
geoffxy@mit.edu

---

*All authors contributed equally to this paper.

Reading your paper: abstract

ABSTRACT

Query optimization is one of the most challenging problems in database systems. Despite the progress made over the past decades, query optimizers remain extremely complex components that require a great deal of hand-tuning for specific workloads and datasets. Motivated by this shortcoming and inspired by recent advances in applying machine learning to data management challenges, we introduce *Neo* (*Neural Optimizer*), a novel learning-based query optimizer that relies on deep neural networks to generate query executions plans. Neo bootstraps its query optimization model from existing optimizers and continues to learn from incoming queries, building upon its successes and learning from its failures. Furthermore, Neo naturally adapts to underlying data patterns and is robust to estimation errors. Experimental results demonstrate that Neo, even when bootstrapped from a simple optimizer like PostgreSQL, can learn a model that offers similar performance to state-of-the-art commercial optimizers, and in some cases even surpass them.

Reading your paper: abstract

ABSTRACT

Query optimization is one of the most challenging problems in database systems. Despite the progress made over the past decades, query optimizers remain extremely complex components that require a great deal of hand-tuning for specific workloads and datasets. Motivated by this shortcoming and inspired by recent advances in applying machine learning to data management challenges, we introduce *Neo* (*Neural Optimizer*), a novel learning-based query optimizer that relies on deep neural networks to generate query executions plans. Neo bootstraps its query optimization model from existing optimizers and continues to learn from incoming queries, building upon its successes and learning from its failures. Furthermore, Neo naturally adapts to underlying data patterns and is robust to estimation errors. Experimental results demonstrate that Neo, even when bootstrapped from a simple optimizer like PostgreSQL, can learn a model that offers similar performance to state-of-the-art commercial optimizers, and in some cases even surpass them.

Statement of the problem / motivation

Reading your paper: abstract

ABSTRACT

Query optimization is one of the most challenging problems in database systems. Despite the progress made over the past decades, query optimizers remain extremely complex components that require a great deal of hand-tuning for specific workloads and datasets. Motivated by this shortcoming and inspired by recent advances in applying machine learning to data management challenges, we introduce *Neo* (*Neural Optimizer*), a novel learning-based query optimizer that relies on deep neural networks to generate query executions plans. Neo bootstraps its query optimization model from existing optimizers and continues to learn from incoming queries, building upon its successes and learning from its failures. Furthermore, Neo naturally adapts to underlying data patterns and is robust to estimation errors. Experimental results demonstrate that Neo, even when bootstrapped from a simple optimizer like PostgreSQL, can learn a model that offers similar performance to state-of-the-art commercial optimizers, and in some cases even surpass them.

Statement of the problem / motivation

Statement of the solution / contribution

Reading your paper: abstract

ABSTRACT

Query optimization is one of the most challenging problems in database systems. Despite the progress made over the past decades, query optimizers remain extremely complex components that require a great deal of hand-tuning for specific workloads and datasets. Motivated by this shortcoming and inspired by recent advances in applying machine learning to data management challenges, we introduce *Neo* (*Neural Optimizer*), a novel learning-based query optimizer that relies on deep neural networks to generate query executions plans. Neo bootstraps its query optimization model from existing optimizers and continues to learn from incoming queries, building upon its successes and learning from its failures. Furthermore, Neo naturally adapts to underlying data patterns and is robust to estimation errors. Experimental results demonstrate that Neo, even when bootstrapped from a simple optimizer like PostgreSQL, can learn a model that offers similar performance to state-of-the-art commercial optimizers, and in some cases even surpass them.

Statement of the problem / motivation

Statement of the solution / contribution

Summary of mechanism / intuition
[sometimes skipped :( ]

Reading your paper: abstract

ABSTRACT

Query optimization is one of the most challenging problems in database systems. Despite the progress made over the past decades, query optimizers remain extremely complex components that require a great deal of hand-tuning for specific workloads and datasets. Motivated by this shortcoming and inspired by recent advances in applying machine learning to data management challenges, we introduce *Neo* (*Neural Optimizer*), a novel learning-based query optimizer that relies on deep neural networks to generate query executions plans. Neo bootstraps its query optimization model from existing optimizers and continues to learn from incoming queries, building upon its successes and learning from its failures. Furthermore, Neo naturally adapts to underlying data patterns and is robust to estimation errors. Experimental results demonstrate that Neo, even when bootstrapped from a simple optimizer like PostgreSQL, can learn a model that offers similar performance to state-of-the-art commercial optimizers, and in some cases even surpass them.

Statement of the problem / motivation

Statement of the solution / contribution

Summary of mechanism / intuition
[sometimes skipped :( ]

Experimental highlight / brag
(if applicable)

Reading your paper: sections

- *Almost* every SIGMOD/VLDB paper has these sections:
  - Introduction: most important part of the paper.
    - States the problem, primary solution, summary of the results, and outlines the paper.
  - Related work: great citations to follow for more info
  - Architecture: overview of the entire system, guide to “details” section of the paper
  - Methods: the nitty-gritty of how the paper was developed.
  - Experiments: empirical justification for the work
  - Conclusion: short summary of the introduction

Reading your paper: sections

- *Almost* every SIGMOD/VLDB paper has these sections:
  - **Introduction: most important part of the paper.**
    - States the problem, primary solution, summary of the results, and outlines the paper.
  - Related work: great citations to follow for more info
  - Architecture: overview of the entire system, guide to “details” section of the paper
  - Methods: the nitty-gritty of how the paper was developed.
  - **Experiments: empirical justification for the work**
  - Conclusion: short summary of the introduction

Reading your paper: sections

- *Almost* every SIGMOD/VLDB paper has these sections:
  - **Introduction: most important part of the paper.**
    - States the problem, primary solution, summary of the results, and outlines the paper.
  - Related work: great citations to follow for more info
  - Architecture: overview of the entire system, guide to “details” section of the paper
  - Methods: the nitty-gritty of how the paper was developed.
  - **Experiments: empirical justification for the work**
  - Conclusion: short summary of the introduction

Outline

- Seminar specifics
- How to read a DB paper
  - **The 3 pass method**
  - Murat’s method
- How to give a good seminar talk

Reading your paper, 3 pass

- First pass: general idea
- Second pass: start to grasp the content
- Third pass: deep understanding

https://rm.cab/3pass

First Pass

- Read the title, abstract, and introduction *carefully*
- Read the section and sub-section headings
- Read the related work
  - Mark papers you've read already (helpful later)

https://rm.cab/3pass

First Pass

- Ask yourself the “C”s:
- *Category:* what type of paper is this?
- *Context:* what other papers might be important for understanding this one?
- *Correctness:* does what I’m reading seem plausible?
- *Contributions*: what are the paper’s main contributions?
- *Clarity*: is the paper well-written?

https://rm.cab/3pass

Second Pass

- Read all sections, but ignore details like proofs or algorithm blocks.
- Special attention to figures.
- Critical references you aren’t familiar with?
- After, you should be able to explain the paper with supporting evidence to someone else.
- If things still aren’t clicking, go to bed.

https://rm.cab/3pass

Third Pass

- Generally only do the 3rd pass if you are presenting or reviewing a paper, or if it is critical to your own research
- You want to understand the paper to *reimplement* it.
- Question each assumption
- Ask yourself in each section: how would I have done this myself? Then, look at what the authors did, and try to figure out why there is a difference
- Write down ideas for future work

https://rm.cab/3pass

3 Pass Method

- Often, you won’t read a whole paper.
  - Stop after pass 1 if there’s something clearly wrong, or the paper isn’t relevant to you
  - Stop after pass 2 if the paper is incomprehensible, you don’t have the right background, or the paper isn’t highly relevant

https://rm.cab/3pass

The 3 Pass Method

Pass	Beginner	Experienced
1	10 – 15m	5 - 10m
2	~ 1h	30m – 1h
3	4 – 5h	1 – 1.5h

https://rm.cab/3pass

Outline

- Seminar specifics
- How to read a DB paper
  - The 3 pass method
  - **Murat’s method**
- How to give a good seminar talk

The Murat Method

1) Print it out
2) Ask questions and argue
3) Write a review
4) Fight the paper
5) Sleep on it and come back

https://rm.cab/murat

Print it Out

- Physically mark and touch the paper
- Some studies show writing notes is more effective than typing them
- Easier to write diagrams or mark specific parts of plots

https://rm.cab/murat

Ask questions and argue

- Ask “why” about each definition / claim / stmt
  - “Critical reading”
- Mark parts you don’t understand “WDYM”
- Guess what the next paragraph will be
- Emotionally engage with the paper

https://rm.cab/murat

Write a review

- “If I had more time, I would’ve written a shorter letter.” - Pascal
- Write a summary of each section, then the entire paper.
  - You will find gaps in your knowledge.
- Think about the authors of the paper reading your review

https://rm.cab/murat

Fight with the Paper

- Poke holes in the paper
- What could have been better?
- What could have been *simpler?*
- Did the authors even ask the right question?

https://rm.cab/murat

Sleep on it

- At this point, we've been very negative.
- Let your subconscious mule it over
- Argue fiercely with a paper, then you can truly learn from it

When I truly understand my enemy, understand him well enough to defeat him, then in that very moment I also love him. I think it's impossible to really understand somebody, what they want, what they believe, and not love them the way they love themselves. - OSC, Ender's Game

Outline

- Seminar specifics
- How to read a DB paper
  - The 3 pass method
  - Murat’s method
- **How to give a good seminar talk**

Presenting a Paper

Great ideas are worthless if you keep them to yourself.

- Adapted from a talk by Mitch Cherniak at Brandeis University
- … which was adapted from a talk by Simon Peyton Jones at MSR

Practice makes Perfect

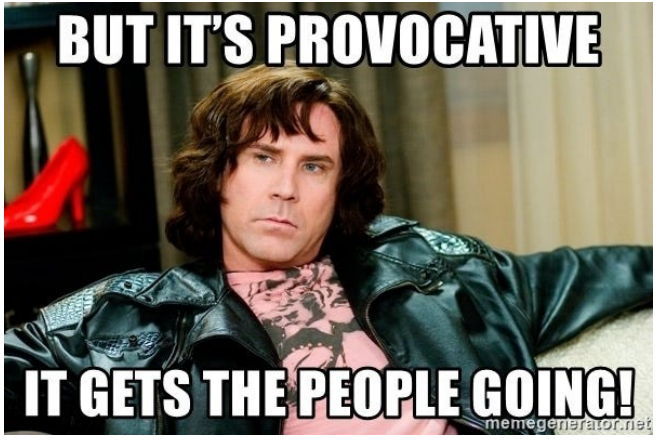
- Giving a good research talk is *the single most important skill* you'll need as a researcher.
- You get good at it the same way you get good at anything.
  - Invest time.
  - Learn skills.
  - Practice.

Purpose of your Talk

- ~~To impress the audien~~
- ~~To give every detail yo~~
- ~~To present every tech~~

Purpose of your Talk

- Give your audience an intuitive feel for an idea
- Make them want to read your paper
- Entertain, engage, and maybe even provoke



You can’t always get the audience you want

Assume your audience

- Has read the paper between 0 and 1 times
- Will not have any background material
- Have a 1:30pm reservation with their significant other’s parents and haven’t gotten changed yet

Your job is to ENGAGE them

What goes in your talk

- Outline
- Motivation
- Key idea
- That’s it

Outline

- Recipe for success:
  - Tell the audience what you are going to do
  - Do the thing
  - Tell the audience what you did (summarize)

Motivation

- You have 2 minutes to engage your audience

  Answer:
- Why should I tune in to this talk?
- What is the problem?
- Why is this interesting?
- Give an example!

The Key Idea

- What is the *one thing* the audience should remember from your talk?
- You must identify the key idea.
  - “Talked about query optimization” is really bad.
- Be specific when you address the main idea.
- Organize your whole talk around this.
  - If it does not relate to the main idea, throw it out.

Examples, examples, examples

- Use examples to…
  - Motivate your work
  - Convey the basic intuition
  - Illustrate your idea in action
  - Show extreme cases
  - Highlight shortcomings

If you are running out of time, omit the general case, not the examples!

What to Leave Out

- Don’t bullshit. (I’m watching)
- It is OK if you don’t understand something.
  - When this happens, omit the details, and show the conclusion.
- Gory details

$$\frac{}{\Gamma \vdash k : \tau_k} \qquad \frac{\Gamma \cup \{x : \tau\} \vdash e : \tau'}{\Gamma \vdash \lambda x.e : \tau \to \tau'} \qquad \frac{\Gamma \vdash e_1 : \mathtt{ST}\ \tau^\circ\ \tau \qquad \Gamma \vdash e_2 : \tau \to \mathtt{ST}\ \tau^\circ\ \tau'}{\Gamma \vdash e_1\ \mathtt{>>=}\ e_2 : \mathtt{ST}\ \tau^\circ\ \tau'}$$

$$\frac{\Gamma \vdash e : \tau}{\Gamma \vdash \mathtt{returnST}\ e : \mathtt{ST}\ \tau^\circ\ \tau} \qquad \frac{\Gamma \vdash e : \tau}{\Gamma \vdash \mathtt{newVar}\ e : \mathtt{ST}\ \tau^\circ\ (\mathtt{MutVar}\ \tau^\circ\ \tau)} \qquad \frac{\Gamma \vdash e : \mathtt{MutVar}\ \tau^\circ\ \tau}{\Gamma \vdash \mathtt{readVar}\ e : \mathtt{ST}\ \tau^\circ\ \tau}$$

$$\frac{\Gamma \vdash e_1 : \mathtt{MutVar}\ \tau^\circ\ \tau \qquad \Gamma \vdash e_2 : \tau}{\Gamma \vdash \mathtt{writeVar}\ e_1\ e_2 : \mathtt{ST}\ \tau^\circ\ \mathtt{Unit}} \qquad \frac{}{\Gamma \cup \{x : \forall \alpha_i.\tau\} \vdash x : \tau[\tau_i/\alpha_i]}$$

$$\frac{\Gamma \vdash e : \tau' \to \tau \qquad \Gamma \vdash e' : \tau'}{\Gamma \vdash e\ e' : \tau} \qquad \frac{\Gamma \vdash e : \mathtt{ST}\ \alpha^\circ\ \tau}{\Gamma \vdash \mathtt{runST}\ e : \tau}\ \alpha^\circ \notin FV(\Gamma, \tau)$$

$$\frac{\forall j.\Gamma \cup \{x_i : \tau_i\}_i \vdash e_j : \tau_j \qquad \Gamma \cup \{x_i : \forall \alpha_{j_i}.\tau_i\}_i \vdash e' : \tau'}{\Gamma \vdash \mathtt{let}\ \{x_i = e_i\}_i\ \mathtt{in}\ e' : \tau'}\ \alpha_{j_i} \in FV(\tau_i) - FV(\Gamma)$$

Figure 1. Typing Rules

What to Leave Out

- Don’t bullshit. (I’m watching)
- It is OK if you don’t understand something.
  - When this happens, omit the details, and show the conclusion.
- Gory details
  - I know you spent hours understanding these.
  - Show (through examples), don’t tell.

What to Leave Out

- (opinion) nobody likes a slide of text

1 week before your talk

- Long summary, show me you understand the paper.
- Come to my office hours, schedule an appointment.
- Practice.

1 hour before your talk

- Trouble breathing, trouble standing, trouble thinking. All totally normal.
- Deep breathing during your practice talks.
- Script the first few sentences of every section
  - No brain required
- Go to the bathroom.
- This happens to everyone.

During your talk

- Be enthusiastic! If you aren’t, why should your audience be?
- Talk to the people in the back
- Make eye contact, find the “nodders”
- Watch for questions

Questions

- Questions are a signal you are doing your job.
- Don’t defer them, encourage them.
- The **absolute best thing** that can happen during your seminar is you run out of time because of too many questions.

The clock is not a suggestion

- With a few minutes left on the clock, no one is paying attention.
- End on time, even if you have to skip material.

Watch others

- Great presenters: Tim Kraska, Leilani Battle, Andy Pavlo, Magdalena Balazinska, Jignesh Patel, Jialin Ding
- DSL seminar, noon to 1pm every Friday

Use your Hint!

Part 2: Rows and Columns		
September 29th	Key question: how much, if any, of the advantages of column stores can we get from a row store? "Column-Stores vs. Row-Stores: How Different Are They Really?" Abadi et al. SIGMOD '08	+ 20/100
October 1st	Key question: how do column stores take advantage of compression? "Integrating compression and execution in column-	+ 19/100

Leading the Discussion

- 1:00pm – 1:30pm
- Come with several questions prepared
  - This should *not* be quiz questions
- Examples:
  - The authors used an tree to locate answers quickly, what about a hash table?
  - What other applications exist for this technique?

Leading the Discussion

- Let folks talk, but try to keep the discussion heading in a fruitful direction
- Link your talk to the speakers before you

Participating in the Discussion

- Ask questions!
- Answer questions asked of you!
- You’ll be up there soon

Summary

- Paper assignments: due Monday at 10am.
  - https://rm.cab/papers
- Long summary: 1 week before your talk
- Short summary: before class each day
- Read your paper: it will take a while
- Give your talk: motivation, main idea, examples.
- Participate in discussion!