Research Statement

Yuhao Zhang
yuz870@eng.ucsd.edu
https://yhzhang.info

Advancements in deep learning have brought workloads that put existing data analytics infrastructures and systems to the test. They ushered in an era of huge workloads that are increasingly computation- and data-intensive. Most run with severe inefficiency and face huge scalability challenges due to suboptimal scheduling, poor resource/memory management, or simply the lack of proper software system support. With the stagnated Moore's law, large-scale and distributed computation is almost inevitable, making all the system issues mentioned more complex. ML System research is born under this context as a cross-section of system and ML research, aiming to optimize and scale up ML workloads.

My research approach. The very core of ML system research is still system research. The various efficiency, cost, and scalability challenges are often the re-discovering of or largely resembles years-old OS, compiler, scheduling, and data system problems. What sets ML workloads aside is usually the vast amount of dense data, huge computational costs, and heavy communications. The main challenge is, thus, often about something other than imagining brand new technologies out of thin air, but rather about correctly identifying the challenges and bottlenecks of ML workloads, then repurposing and innovating upon well-established system techniques. The core mission is: given the rather finite set of techniques, how do we synergize and innovate upon them to accommodate the ever-changing and seemly infinite variations of ML/AI workloads?

My research goal and impact. My primary research goal is to provide software system support for very diversified ML workloads to make data science faster and easier. In my past research and ongoing research, I captured the core challenges of various ML workloads on different data modalities, ranging from: model selection and training workloads on i.i.d. (tabular and images) data, both on data lake data [2] and data within data management systems [5]; Unbounded vocabulary querying workloads on video and image data [4]; And Graph Neural Network model selection and training workloads on graph data [3]. I proposed novel techniques and built systems to boost the throughput and increase scalability. All of my previous work is released open-sourcedly. My past research of Cerebro [2] and Cerebro-DS [5] have been incorporated into the Apache MADlib open-source project and offered in Greenplum Database by VMware. Databricks are also reviewing the same project to offer to their customers. Graph DBMS vendor is also interested in my work of Lotan [3].

Past Research

Distributed system for model selection workloads on i.i.d. (tabular and images) data. There is a major bottleneck to the wider adoption of deep learning: the pain and resource intensiveness of model selection. It is an empirical process this involves exploring deep net architectures and hyper-parameters, often requiring hundreds of trials. However, most ML systems focus on training one model at a time, reducing throughput and raising overall resource costs; some also sacrifice reproducibility. Towards higher throughput and resource utilization and as a part of a grander vision [1] of deep learning model selection system, I built Cerebro [2] with my advisor and college. Cerebro is a data system to raise deep net model selection throughput at scale without raising resource costs or sacrificing reproducibility or accuracy. Cerebro uses a new parallel deep learning training strategy called model hopper parallelism. It hybridizes task- and data-parallelism to mitigate the cons of these prior paradigms and offer the best of both worlds. Experiments on large ML benchmark datasets showed that Cerebro offers 3x to 10x runtime savings relative to data-parallel systems like Horovod and Parameter Server and up to 8x memory/storage savings or up to 100x network savings relative to task-parallel systems. Cerebro also supports heterogeneous resources and fault tolerance.

Bringing deep learning to data system-resident data. Deep learning's popularity is not only limited to ML researchers; many enterprises and businesses are also considering adopting deep learning for their data analytics applications. Large business-critical datasets in such settings typically reside in RDBMSs or other data systems. In the work of Cerebro above, I explored the landscape of standalone deep learning model selection systems and proposed a new parallelism. However, it was unclear if the parallelism and scheduling could be incorporated into existing infraustuctures of data management systems. In the project of Cerebro-DS [5], I characterized the particular suitability of Cerebro on data systems. However, to bring the novel model hopper parallelism to DB resident data, I showed that there was no single "best" approach, and an exciting tradeoff space of approaches exists. I explained four canonical approaches, built prototypes upon Greenplum Database, and compared them analytically on multiple criteria (e.g., runtime efficiency and ease of governance) with large-scale DL workloads. The experiments and analyses showed that it was non-trivial to meet all practical desiderata well, and there was a Pareto frontier; for instance, some approaches are 3x-6x faster but fare worse on governance and portability. These results and insights can help DBMS and cloud vendors design better DL support for DB users.

System for Graph Neural Network model selection and training workloads on graph data. Moving on from the common i.i.d. data and models designed for them, I looked at the rapidly developing Graph Neural Networks on graph data. The complexity of GNN training and the challenges of scalability has also sparked interest from the ML system community, with efforts to build systems that provide higher efficiency, better memory management, and schemes to reduce costs. However, many of these systems reinvent the wheel by reiterating years of research and development on advanced graph data systems. Further, they often couple the scalability challenges of graph data processing with those of GNN training, resulting in entangled complex problems and systems that need help to handle either scalability challenge. Lotan [3] is a highly scalable data system for full-batch GNN training with the separation of graph and neural network. Lotan provides separation of the graph and neural network computation, free individual scaling for each, and bridges existing graph data systems and deep learning frameworks. Lotan offers a series of technical innovations, including execution plan rewriting, highly-efficient data movement between systems, a GNN-centric graph partitioning method and the corresponding gradients backpropagation scheme, and GNN model batching. By taking several different GNN workloads, I demonstrated the system's capability of training large GNN models that prior art fails to handle. The system can surpass the state-of-art systems' training throughput by over 40x and can beat a naively implemented in-data-system GNN training framework by 76x. I believe the system can increase efficiency for existing workloads and open new possibilities for future GNN algorithmic research.

System for unbounded vocabulary querying workloads on video and image data Video data is sometimes dubbed as "fast data"", characterized by their sheer volume and the requirement of real-time responses for many applications such as video monitoring. Using deep learning methods for these applications incurs high computational costs and inference latency. The prior art has studied how to improve system efficiency. Nevertheless, they largely focus on small "closed world" prediction vocabularies, even though many surveillance security and traffic analytics applications have an ever-growing set of target entities. I call this the "unbounded vocabulary" issue, which is a key bottleneck for emerging video monitoring applications. I presented the first data system for tacking this the issue for video querying, Panorama [4]. The design philosophy is to build a unified and domain-agnostic system that lets application users generalize to unbounded vocabularies in an out-of-the-box manner without tedious manual re-training. To this end, I synthesized and innovated upon an array of techniques from the ML, vision, databases, and multimedia systems literature to devise a new system architecture. I also presented techniques to ensure Panorama had high inference efficiency. Experiments with multiple real-world datasets showed that Panorama could achieve between 2x to 20x higher efficiency than baseline approaches on in-vocabulary queries while still yielding comparable accuracy and also generalizing well to unbounded vocabularies.

Future Research

ML System research is a relatively new domain, and vast opportunities exist. ML-based apps will be ubiquitous eventually, and there are destined to be new challenges. It is still the right time to do ML system research. We are at the transition point from segmented and often ad-hoc solutions to fully-fledged systems that will form the backbones of the next wave of technological innovation. I choose to conduct future research aligned with my current experiences and remain at the frontline during this postdoc.

From homogenous to heterogenous ML systems. As ML/AI research grows in complexity. Because the world is heterogeneous and multimodal, highly heterogenous environments will only become more common, where distinct ML model architectures, multiple data storage, diversified data modalities, and various computational resources are all involved. Such complexities must be abstracted away from the user for easy adoption. I imagine a poly-ML system like the polystore systems in data system research. There are many system problems to expect: first, logical plan optimization of multiple correlated workloads. Second, scheduling and coordination of heterogenous computational graph operators. Third, resource management of heterogenous physical clusters. Can we build a platform to facilitate multimodal ML research? Can we develop a set of abstractions and logical operators to express most multimodal machine learning research? Can we allow the user to easily manipulate the data and experiment with their models without worrying about the underlying heterogeneous computational parallelism, data storage, and physical resources?

From ML frameworks to distributed ML data systems. The future of computation is distributed, as the performance of a single machine is physically limited, while distributed processing has much more room to scale. However, today's practitioners still predominantly use a single machine, although they are offered data-, model-, and other more advanced parallelisms. The adoption of these parallel techniques is not ideal, despite the huge amount of effort made. One of the major culprits behind the under-utilization of distributed processing is the systems' common lack of support for distributed storage, querying, and data computation. A common pattern is the ML system trying to parallelize and rewrite the users' single-node script. Instead of writing a single-node ML program with single-node operators and then trying to compile and distributedly execute it, can we offer distributed counterparts of the ML operators directly to the user? In the world of SQL and MapReduce, the users do not write a script and execute it on every machine; they code directly in a declarative way so that their program is readily parallelized. Instead of throwing large deep learning matrix operations to a single GPU and failing when there are memory limitations, can we naturally chunk all of the matrices and distribute them across the cluster, and gracefully swap GPU memory to DRAM and potentially to disk like data systems do?

From optimizing costs to enabling innovations. So far, ML system research has been primarily developed in parallel with ML algorithm research. The trend is shifting to some extent, but in general, it was the system researchers trying to catch up with the latest innovations from the ML community and building systems for their emerging use cases. This pattern usually results in a lag behind the frontline of ML research. Consequently, most ML system research is only designed to optimize costs and provide better usability for existing models and workloads. However, in an era of fast-paced innovations, it is hard to predict what comes next, and the present challenges may be rendered trivial by the next generation of models and workloads. To get ahead of this curve, I aspire to build truly future-proof systems that can not only "democratize ML" by optimizing the costs but also has the potential to enable impossible workloads and model architectures and stimulate ML algorithmic innovations. I look forward to collaborating with ML researchers and enabling model architectures that were not possible with limited computational resources. On the other hand, there are opportunities to design ML system-aware models. Can we build model architectures designed so that their data dependencies, data access, and communication patterns are easy to parallelize and well-optimized by the underlying system? The model can even assume data locality, cache, and sparsity. Can we abstract and summarize a set of principles about scalable model architecture design? Can we build model architecture search systems that, besides accuracy, also optimize for runtime efficiency and scalability?

References

[1] A. Kumar, S. Nakandala, **Yuhao Zhang**, S. Li, A. Gemawat, and K. Nagrecha. (Vision paper) Cerebro: A Layered Data Platform for Scalable Deep Learning. In *CIDR*. www.cidrdb.org, 2021.

[2] S. Nakandala, **Yuhao Zhang**, and A. Kumar. Cerebro: A Data System for Optimized Deep Learning Model Selection. *Proc. VLDB Endow.*, 13(11):2159–2173, 2020.

[3] **Yuhao Zhang** and A. Kumar. Lotan: A Highly Scalable System for GNN Training via the Separation of Graph and Neural Network. *Under Submission*.

[4] **Yuhao Zhang** and A. Kumar. Panorama: A data system for unbounded vocabulary querying over video. *Proc. VLDB Endow.*, 13(4):477–491, 2019.

[5] **Yuhao Zhang**, F. Mcquillan, N. Jayaram, N. Kak, E. Khanna, O. Kislal, D. Valdano, and A. Kumar. Distributed Deep Learning on Data Systems: A Comparative Analysis of Approaches. *Proc. VLDB Endow.*, 14(10):1769–1782, 2021.