SysML: The New Frontier of Machine Learning Systems

Alexander Ratner[1] Dan Alistarh[2] Gustavo Alonso[3] David G. Andersen[4,5] Peter Bailis[1,6] Sarah Bird[7] Nicholas Carlini[5] Bryan Catanzaro[8] Jennifer Chayes[7] Eric Chung[7] Bill Dally[1,8] Jeff Dean[5] Inderjit S. Dhillon[9,10] Alexandros Dimakis[9] Pradeep Dubey[11] Charles Elkan[12] Grigori Fursin[13,14] Gregory R. Ganger[4] Lise Getoor[15] Phillip B. Gibbons[4] Garth A. Gibson[16,17,4] Joseph E. Gonzalez[18] Justin Gottschlich[11] Song Han[19] Kim Hazelwood[20] Furong Huang[21] Martin Jaggi[22] Kevin Jamieson[23] Michael I. Jordan[18] Gauri Joshi[4] Rania Khalaf[24] Jason Knight[11] Jakub Konečný[5] Tim Kraska[19] Arun Kumar[12] Anastasios Kyrillidis[25] Aparna Lakshmiratan[20] Jing Li [26] Samuel Madden[19] H. Brendan McMahan[5] Erik Meijer[20] Ioannis Mitliagkas[27,28] Rajat Monga[5] Derek Murray[5] Kunle Olukotun[1,29] Dimitris Papailiopoulos[26] Gennady Pekhimenko[30] Christopher Ré[1] Theodoros Rekatsinas[26] Afshin Rostamizadeh[5] Christopher De Sa[31] Hanie Sedghi[5] Siddhartha Sen[7] Virginia Smith[4] Alex Smola[10,4] Dawn Song[18] Evan Sparks[32] Ion Stoica[18] Vivienne Sze[19] Madeleine Udell[31] Joaquin Vanschoren[33] Shivaram Venkataraman[26] Rashmi Vinayak[4] Markus Weimer[7] Andrew Gordon Wilson[31] Eric Xing[4,34] Matei Zaharia[1,35] Ce Zhang[3] Ameet Talwalkar*[4,32]

[1]Stanford, [2]IST Austria, [3]ETH Zurich, [4]Carnegie Mellon University, [5]Google, [6]Sisu Data, [7]Microsoft, [8]NVIDIA, [9]University of Texas at Austin, [10]Amazon, [11]Intel, [12]University of California San Diego, [13]cTuning Foundation, [14]Dividiti, [15]UC Santa Cruz, [16]Vector Institute, [17]Univerrsity of Toronto, [18]UC Berkeley, [19]MIT, [20]Facebook, [21]University of Maryland, [22]EPFL, [23]University of Washington, [24]IBM Research, [25]Rice University, [26]University of Wisconsin-Madison, [27]Mila, [28]University of Montreal, [29]SambaNova Systems, [30]University of Toronto, [31]Cornell University, [32]Determined AI, [33]Eindhoven University of Technology, [34]Petuum, [35]Databricks

April 1, 2019

Abstract

Machine learning (ML) techniques are enjoying rapidly increasing adoption. However, designing and implementing the systems that support ML models in real-world deployments remains a significant obstacle, in large part due to the radically different development and deployment profile of modern ML methods, and the range of practical concerns that come with broader adoption. We propose to foster a new systems machine learning research community at the intersection of the traditional systems and ML communities, focused on topics such as hardware systems for ML, software systems for ML, and ML optimized for metrics beyond predictive accuracy. To do this, we describe a new conference, SysML, that explicitly targets research at the intersection of systems and machine learning with a program committee split evenly between experts in systems and ML, and an explicit focus on topics at the intersection of the two.

1 Introduction

Over the last few years, machine learning (ML) has hit an inflection point in terms of adoption and results. Large corporations have invested billions of dollars in reinventing themselves as "AI-centric"; swaths of academic disciplines have flocked to incorporate machine learning into their research; and a wave of excitement about AI and ML has proliferated through the broader public sphere. This has been due to several factors, central amongst them new deep learning approaches, increasing amounts of data and compute resources, and collective investment in open-source frameworks like Caffe, Theano, MXNet, TensorFlow, and PyTorch, which have effectively decoupled model design and specification from the systems to implement these models. The resulting wave of technical advances and practical results seems poised to transform ML from a bespoke solution used on certain narrowly-defined tasks, to a commodity technology deployed nearly everywhere.

Unfortunately, while it is easier than ever to run state-of-the-art ML models on pre-packaged datasets, designing and implementing the *systems* that support ML in real-world applications is increasingly a major bottleneck. In large part this is because ML-based applications require distinctly new types of software, hardware, and engineering systems to support them. Indeed, modern ML applications have been referred to by some as a new "Software 2.0" [5] to emphasize the radical shift they represent as compared to traditional computing applications. They

*Corresponding author, talwalkar@cmu.edu.

are increasingly *developed* in different ways than traditional software—for example, by collecting, preprocessing, labeling, and reshaping training datasets rather than writing code—and also *deployed* in different ways, for example utilizing specialized hardware, new types of quality assurance methods, and new end-to-end workflows. This shift opens up exciting research challenges and opportunities around high-level interfaces for ML development, low-level systems for executing ML models, and interfaces for embedding learned components in the middle of traditional computer systems code.

Modern ML approaches also require new solutions for the set of concerns that naturally arise as these techniques gain broader usage in diverse real-world settings. These include *cost* and other efficiency metrics for small and large organizations alike, including e.g. computational cost at training and prediction time, engineering cost, and cost of errors in real-world settings; *accessibility* and *automation*, for the expanding set of ML users that do not have PhDs in machine learning, or PhD time scales to invest; *latency* and other run-time constraints, for a widening range of computational deployment environments; and concerns like *fairness, bias, robustness, security, privacy, interpretability, and causality*, which arise as ML starts to be applied to critical settings where impactful human interactions are involved, like driving, medicine, finance, and law enforcement.

This combination of radically different application requirements, increasingly-prevalent systems-level concerns, and a rising tide of interest and adoption, collectively point to the need for a concerted research focus on the systems aspects of machine learning. To accelerate these research efforts, our goal is to help foster a new *systems machine learning community* dedicated to these issues. We envision focusing on broad, full-stack questions that are complementary to those traditionally tackled independently by the ML and Systems communities, including:

1. **How should software systems be designed to support the full machine learning lifecycle, from programming interfaces and data preprocessing to output interpretation, debugging and monitoring? Example questions include:**
   - *How can we enable users to quickly "program" the modern machine learning stack through emerging interfaces such as manipulating or labeling training data, imposing simple priors or constraints, or defining loss functions?*
   - *How can we enable developers to define and measure ML models, architectures, and systems in higher-level ways?*
   - *How can we support efficient development, monitoring, interpretation, debugging, adaptation, tuning, and overall maintenance of production ML applications- including not just models, but the data, features, labels, and other inputs that define them?*
2. **How should hardware systems be designed for machine learning? Example questions include:**
   - *How can we develop specialized, heterogeneous hardware for training and deploying machine learning models, fit to their new operation sets and data access patterns?*
   - *How can we take advantage of the stochastic nature of ML workloads to discover new trade-offs with respect to precision, stability, fidelity, and more?*
   - *How should distributed systems be designed to support ML training and serving?*
3. **How should machine learning systems be designed to satisfy metrics beyond predictive accuracy, such as power and memory efficiency, accessibility, cost, latency, privacy, security, fairness, and interpretability? Example questions include:**
   - *How can machine learning algorithms and systems be designed for device constraints such as power, latency, and memory limits?*
   - *How can ML systems be designed to support full-stack privacy and security guarantees, including, e.g., federated learning and other similar settings?*
   - *How can we increase the accessibility of ML, to empower an increasingly broad range of users who may be neither ML nor systems experts?*

Another way of partitioning these research topics is into **high-level systems** for ML that support interfaces and workflows for ML development—the analogue of traditional work on programming languages and software engineering—and **low-level systems** for ML that involve hardware or software—and that often blur the lines between the two—to support training and execution of models, the analogue of traditional work on compilers and architecture. Regardless of the ontology, we envision these questions being addressed by a strong mix of theoretical, empirical, and applications-driven perspectives. And given their full-stack nature, we see them being best answered by a research community that mixes perspectives from the traditional machine learning and systems communities.

A separate but closely related and increasingly exciting area of focus is *machine learning for systems*: the idea of applying machine learning techniques to improve traditional computing systems. Examples include replacing the data structures, heuristics, or hand-tuned parameters used in low-level systems like operating systems, compilers, and storage systems with learned models. While this is clearly a distinct research direction, we also see the systems machine learning community as an ideal one to examine and support this line of work, given the required confluence of ML and systems expertise.

Finally, we see the systems machine learning community as an ideal jumping-off point for even larger-scale and broader questions, beyond how to interface with, train, execute, or evaluate single models [4]. For instance, how do we manage entire ecosystems of models that interact in complex ways? How do we maintain and evaluate systems that pursue long term goals? How do we measure the effect of ML systems on societies, markets, and more? How do we share and reuse data and models at societal scale, while maintaining privacy and other economic, social, and legal issues? All of these questions and many more will likely need to be approached by research at the intersection of traditional machine learning and systems viewpoints.

2 Why Now? The Rise of Full Stack Bottlenecks in ML

Researchers have worked at the intersection of systems and machine learning research for years—but this *systems ML* work has moved to the forefront recently due to leaps in machine learning performance on challenging benchmark tasks, and the growing realization that a range of new systems up and down the computing stack are needed to translate this academic promise to real-world practice [7; 6].

In recent years, often driven by new deep learning approaches, the field of machine learning has made significant leaps forward on benchmark tasks in traditional 'grand challenge' domains like image classification, text and speech processing, and others. In certain benchmarks, ML models have even surpassed human performance [3]. However, in real-world deployments, a range of bottlenecks begin to surface, which crucially are full-stack, systems-level concerns, rather than solely properties of the core machine learning algorithms. These include:

- **Deployment concerns:** As ML becomes used in increasingly diverse and mission-critical ways, a new crop of systems-wide concerns has become increasingly prevalent. These include robustness to adversarial influences or other spurious factors; safety more broadly considered; privacy and security, especially as sensitive data is increasingly used; interpretability, as is increasingly both legally and operationally required; fairness, as ML algorithms begin to have major effects on our everyday lives; and many other similar concerns.
- **Cost:** The original default solution for learning a CNN over ImageNet cost $2,300 of compute and took 13 days to train [2]. Annotation of the huge volumes of training data can alone cost hundreds of thousands to millions of dollars. Reducing cost measured in terms of other metrics—such as latency or power—is also critical for an expanding range of device and production deployment profiles.
- **Accessibility:** As a widening range of people rush to use ML for actual production purposes—including a new breed of polyglot data scientists trained by large new programs at universities—ML systems need to be usable by developers and organizations without PhD-level machine learning and systems expertise.

The shared element in these emerging pain points is that they are full-stack issues that require reasoning not just over core ML algorithms and methods, but the hardware, software, and overall systems that support them as well. As ML adoption and excitement increases, full-stack solutions to the above pain points will be increasingly critical to bridging the gap between machine learning's promise and real-world utility.

3 SysML: Building a New Conference at the Intersection of Systems + Machine Learning

In our view, these fundamental gaps between ML's current promise and its actual usability in practice all concern questions at the intersection of the traditional systems and machine learning communities. As such, we have created a new conference, called the Conference on Systems and Machine Learning (SysML), to target research at the intersection of systems and machine learning, and with the hope of providing an intellectual space that fosters interdisciplinary research that cuts across both. The conference aims to elicit new connections amongst these fields, including identifying best practices and design principles for machine learning systems, as well as developing novel learning methods and theory tailored to practical machine learning workflows.

SysML builds on the dramatic success and growth of satellite workshops connected to leading ML and Systems conferences, e.g., at NeurIPS, ICML, OSDI, and SIGMOD. Following the standards of top conferences in both fields, the SysML review process is a rigorous, highly selective process. However, unlike traditional ML or Systems conferences, the SysML Program Committee consists of experts from both ML and Systems who review all papers together, and has an explicit focus on topics that fall in the interdisciplinary SysML space (as opposed to the broad ML or broad Systems space). Finally, to spur reproducibility and rapid progress in this research area, SysML embraces modern artifact evaluation processes that have been successful at other conferences [1].

SysML was established in 2018 by the inaugural Organizing Committee (Peter Bailis, Sarah Bird, Dimitris Papailiopoulos, Chris Ré, Ben Recht, Virginia Smith, Ameet Talwalkar, Matei Zaharia), led by Program Chair Ameet Talwalkar and Co-Chair Dimitris Papailiopoulos, and with guidance from the Steering and Program Committees.

- *Steering Committee:* Jennifer Chayes, Bill Dally, Jeff Dean, Michael I. Jordan, Yann LeCun, Fei-Fei Li, Alex Smola, Dawn Song, Eric Xing.
- *Program Committee:* David Andersen, Bryan Catanzaro, Eric Chung, Christopher De Sa, Inderjit Dhillon, Alex Dimakis, Charles Elkan, Greg Ganger, Lise Getoor, Phillip Gibbons, Garth Gibson, Joseph Gonzalez, Furong Huang, Kevin Jamieson, Yangqing Jia, Rania Khalaf, Jason Knight, Tim Kraska, Aparna Lakshmiratan, Samuel Madden, Brendan McMahan, Ioannis Mitliagkas, Rajat Monga, Derek Murray, Kunle Olukotun, Theodoros Rekatsinas, Afshin Rostamizadeh, Siddhartha Sen, Evan Sparks, Ion Stoica, Shivaram Venkataraman, Rashmi Vinayak, Markus Weimer, Ce Zhang.

4 Conclusion

There is an incredibly exciting set of research challenges that can be uniquely tackled at the intersection of traditional machine learning and systems communities, both today and moving forward. Solving these challenges will require advances in theory, algorithms, software, and hardware, and will lead to exciting new low-level systems for executing ML algorithms, high-level systems for specifying, monitoring, and interacting with them, and beyond that, new paradigms and frameworks that shape how machine learning interacts with society in general. We envision the new SysML conference as a center of research in these increasingly important areas.

References

[1] ACM: Artifact reviewing and badging. `https://www.acm.org/publications/policies/artifact-review-badging`, 2018.

[2] C. Coleman, D. Narayanan, D. Kang, T. Zhao, J. Zhang, L. Nardi, P. Bailis, K. Olukotun, C. Ré, and M. Zaharia. DAWNBench: An end-to-end deep learning benchmark and competition. *NeurIPS ML Systems Workshop*, 2017.

[3] P. Eckersley, Y. Nasser, et al. EFF AI progress measurement project. `https://eff.org/ai/metrics`, 2017.

[4] M. Jordan. Artificial intelligence—the revolution hasn't happened yet. 2018. `https://medium.com/@mijordan3/artificial-intelligence-the-revolution-hasnt-happened-yet-5e1d5812e1e7`.

[5] A. Karpathy. Software 2.0. `https://medium.com/@karpathy/software-2-0-a64152b37c35`, 2017.

[6] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, and M. Young. Machine learning: The high interest credit card of technical debt. 2014.

[7] A. Talwalkar. Toward the jet age of machine learning. *O'Reilly*, 2018. `https://www.oreilly.com/ideas/toward-the-jet-age-of-machine-learning`.