Introduction to the Special Section: Launching an Agenda for Research on Learning Machine Learning

R. BENJAMIN SHAPIRO, Department of Computer Science, University of Colorado Boulder

REBECCA FIEBRINK, Department of Computing, Goldsmiths University of London

THE EXCITING AND FRAUGHT DAWN OF UBIQUITOUS MACHINE LEARNING

Machine Learning (ML) is exploding in popularity. What little more than a decade ago was the realm of specialists is now becoming a commonplace tool in the development of technology products. For example, as of January 2019, Amazon has sold more than 100 million devices running its Alexa software—whose speech synthesis, recognition, and dialogue systems are all built with ML [Bohn 2019].

Alongside this growth in the application of ML, universities are investing massive resources in hiring faculty who are experts in artificial intelligence, ML, or data science, with approximately 35% of recent faculty searches in top 100 Ph.D.-granting computing departments targeting such faculty [Wills 2019]. These faculty, in turn, are teaching an ever-larger number of ML courses, thereby preparing a wave of software engineers to use ML in their work. These in-person university courses are complemented by online ML education offerings from Coursera, Udacity, EdX, and others that tout enrollments in the millions. ML-focused industry job advertisements are also rapidly growing in number. The number of industry job postings mentioning ML or artificial intelligence on Indeed rose about 30% between 2018 and 2019, after double- and triple-digit growth in the preceding 2 years [Indeed Editorial Team 2019].

This rapid growth in ML-based products, ML faculty positions, and ML engineering jobs is propelled by one basic phenomenon: ML is capable of leveraging ever-increasing amounts of data to inform computational tasks such as decision making, reasoning, and even design and creation of new systems. This means that systems created with ML can be far better than those created with traditional programming approaches at capturing the nuances of many phenomena that matter to people: human bodies, human communication, human societies, and the natural and human-built physical world. Some examples illustrate this point:

- In medicine, ML is finding use in applications from automated diagnosis to flagging patients at risk of postsurgical complications to informing the definitions of diseases and their subtypes [Saria et al. 2018].
- Improvements in ML have driven dramatic increases in the accuracy of natural language translation. For instance, switching to a deep learning-based translation algorithm improved Google's translation accuracy between English and Spanish, French, and Chinese by 60% [Wu et al. 2016].

Authors' addresses: R. B. Shapiro, Department of Computer Science University of Colorado Boulder; email: ben.shapiro@colorado.edu; R. Fiebrink, Department of Computing Goldsmiths University of London; email: r.fiebrink@gold.ac.uk.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2019 Copyright held by the owner/author(s).

1946-6226/2019/10-ART30

<https://doi.org/10.1145/3354136>

- ML underpins content recommendation systems used daily by millions of people for tasks as diverse as music listening [Jacobson et al. 2016], job searching [Sharma and Yan 2013], and online dating [Kircher 2016].
- ML is driving advances in numerous subfields of computer science (CS) as well, from computer vision [Krizhevsky et al. 2012] to computer architecture [Dean et al. 2018; Jiménez and Lin 2002].

Many of the impacts of ML are positive. Technologies that aid in the correct diagnosis of disease are surely a boon to society, as are language translation technologies that facilitate dialogue across cultures. But some impacts are negative. Social media companies' reactive approaches to community management left them utterly unprepared to handle the negative impacts of their click-driven advertising optimization on voters' knowledge during the 2016 election [Tromble and McGregor 2019; Starbird 2019]. Amazon recently had to disable an internal, automated job application triage system that learned, based on data about Amazon's historical hiring practices, to penalize applicants whose resumes included the text "women's," as might be seen on the resumes of people who have enrolled in women's studies programs or played women's hockey (Cowgill and Tucker, Forthcoming). As nearly all such people would be women, the system in effect learned to practice gender discrimination. Similarly problematic systems have been deployed in mortgage lending, prison sentencing, and other domains where they can substantially impact people's lives and livelihoods [O'Neil 2016].

Thus, ML is a double-edged sword—one whose mastery can permit software engineers to create systems that are incredibly beneficial or terribly harmful. The Janus-like nature of new technologies is not novel to the advent of ML (text messaging permits distanced conveyance of affection to loved ones, as well as cyberbullying; computation facilitated humans walking on the moon within decades of IBM facilitating the Nazi-perpetrated holocaust). But the extent to which ML-based systems may reinscribe past inequity in new computational systems creates an especially acute moment to reinvest in ethics education for computing students. It also affords us new opportunities to reassess how computing education can integrate technical and humanistic perspectives including not just ethics but also user experience design, inclusive design, and the creative arts.

THE NEED FOR ML EDUCATION RESEARCH

Despite the investments in ML mentioned earlier, we know remarkably little about how people learn to create, evaluate, and improve ML-based systems. At the time we proposed this special issue, we were unable to locate any peer-reviewed research on the topic of ML education, and very little has been published since. So the impetus for this special issue was to create the first designated venue for research on the teaching, learning, and understanding of ML. Researchers, particularly early career researchers, often need to be strategic about the work they do: if there is no clear venue to publish work from one possible line of study, then they may choose to focus their attention elsewhere. In proposing and then editing this issue, we have sought to create such a venue and thus to catalyze this much needed research. We believe that we have succeeded and hope that many more articles will soon be submitted to TOCE, to the journal *Computer Science Education*, and to conferences like International Computing Education Research (ICER) and the ACM SIGCSE Technical Symposium.

WHAT WE SHOULD LEARN

The advent of widespread ML raises a number of topics about which it would be beneficial for those doing computing education research (CER) to develop an evidentiary base. Some of these topics

are ones for which there is already closely related prior work in CER but where additional study is needed in relation to ML. Other topics are new to CER but have been investigated in some form in other research communities, such as by math or statistics education researchers. Still others are particular enough to ML that they have not, to our prior knowledge, been studied at all. We now enumerate a sampling of these questions that we hope will inspire follow-on work by readers of this article.

What kinds of mathematical, statistical, or computing prerequisites are effective at preparing CS students for mastering ML theory? How should these prerequisites differ for those who wish to apply ML in their work? The application of ML in one's work likely calls for a different set of skills than the development of new ML algorithms, and the population doing the former will likely be orders of magnitude larger than those doing the latter [Simard et al. 2017]. It would be quite helpful for researchers to uncover what topics are useful preparation for these two populations, how these prerequisites overlap, and where they diverge. We note with concern that ACM's and IEEE's current curriculum recommendations [Association for Computing Machinery and IEEE Computing Society 2013] explicitly recommend against CS departments requiring courses in probability and statistics ("we still believe it is not necessary for all CS programs to require a full course in probability theory for all majors."). Because as applied ML is a domain of applied probability and statistics, we expect that some preparation in these subjects is likely to be helpful, although researchers have not yet determined how much or what kind of preparation outside of the context of applying ML will tend to be valuable. Discovering this is an important topic for ML education research, and one that offers considerable opportunity for collaboration with statistics education researchers as well as for exploitation of the knowledge that those researchers have already generated.

Applied ML is an empirical practice; the quality of ML models is a function of the quality of data used to create them. The quality of those data is, in turn, a function of how well ML practitioners construct measurements of the ground truth about which models are intended to make inferences, then use those measurements to engineer useful features with which to create training data. *How do students learn to create good measurements of the real world? How do modelers learn to engineer useful features? How do modelers learn to balance tradeoffs between the explicit construction of semantically meaningful features and letting deep learning algorithms construct features for themselves?* As the latter practice can create challenges for model interpretability and explainability, we might also ask this: How do modelers learn to engineer models to perform well while also being explainable or interpretable? As a first step in this area, it might be exceedingly helpful for researchers to conduct cognitive clinical interviews [Clement 2000; Sherin 2013] or cognitive ethnographies [Hutchins 1995] of experts at work, both of which could offer insights into what being an expert in this kind of work entails.

Another aspect of applied ML's intrinsically empirical nature is the vital importance of learning to evaluate models. It is common for ML practitioners to use techniques such as cross validation, confusion matrices, and ROC curves to evaluate the overall performance of models. *How do modelers interpret the results of such techniques? How do they interpret textual and graphical representations of models' performance?* Researchers might investigate how modelers learn to compare ML classifiers' post hoc distributions with the distribution of labels in a supervised or semisupervised training set. And then they might study how modelers learn to draw productive inferences about the next steps they could take to improve model performance. These are essential ML modeling practices, but no published work describes expert or novice performances of these tasks. Those descriptions would be incredibly valuable research contributions.

How do students conceptualize and reason about the particularities of different ML algorithms and parameters for those algorithms? For example, how do students understand, and develop more productive understandings of, different neural network activation functions and architectures kernels

for support vector machines? This is a topic that in many ways is particular to the domain of ML and thus may be less supportive of building connections to other research disciplines than others we cover in this section. But it is not so unique that ML education researchers cannot be informed by work that has already been done in other fields. It may be helpful to adopt perspectives from the learning sciences on what knowledge is, how it is structured, how it is applied, how it is refined over time, and how all of these things can be studied [Smith et al. 1994; Russ and Odden 2017].

What do tomorrow's software engineers need to know to work with hybrid code+data systems, and what are the best ways to help them to learn it? ML models are typically embedded in software systems that are programmed by humans. These systems can be understood as compositions of functions that are defined through statistical (ML) or symbolic (code) means. The pragmatics of constructing, testing, and maintaining these systems raise numerous research questions related to software engineering education. Just as educators and researchers today puzzle over how best to help students learn to use version control systems [Haaranen and Lehtinen 2015] and how to write good test cases [Politz et al. 2014], research tomorrow might examine how students learn to manage ever-growing amounts of data and to test systems whose behavior may change because of new patterns in those data. Further, research might examine how to support students in reasoning about when ML is likely to be useful and when human programming is likely to be more appropriate.

What kinds of preparation are effective in helping budding engineers to make ethical choices, and to actively resist unethical directions, when they are in the workplace? Ethics education is not a new subfield within computing education, but the collection of ever-larger repositories of personal data combined with increasing computational mediation of basic social and societal functions, and the public unaccountability of large technology companies (e.g., Facebook) does raise new challenges. We do not yet know what combination of basic philosophical education about ethics and analysis of case studies of ethical decision making within technology engineering will best prepare students to analyze new problems before them. Nor do we know much about how to prepare students to effectively resist unethical conduct within technology companies. Computing educators, and computing education researchers, may find much knowledge to draw on within the fields of civics education, anthropology [Goodwin 1994], sociology, sociotechnical systems, and even studies of effective labor organizing.

What do tomorrow's practitioners in other domains—across the sciences, industry, arts, and humanities—need to know about ML, and what do effective approaches to teaching and learning look like across these diverse domains? As the reach of ML spreads, non-computer scientists will increasingly rely on ML to support decision making and deepen understanding of data relevant to their own pursuits. What knowledge will enable such people to use ML effectively and wisely, or enable them to work more effectively with data scientists and ML experts within their organizations? Recent research (e.g., reviewed in Dudley and Kristensson [2018]) also demonstrates that—given appropriate algorithms and user interfaces—practitioners in diverse domains are quite capable of designing and training new ML systems themselves to suit their own needs. ML can be a boon to end-user design [Bernardo et al. 2017], enabling nonprogrammers to build new systems by curating data describing desired system behavior and providing it to a suitable ML algorithm. Further, ML offers particular benefits in to the development of new embodied interfaces—for instance, enabling the creation of bespoke interfaces for gestural control for accessibility and the arts [Zimmermann-Niefield Shapiro and Kane 2019; Fiebrink et al. 2011]—because it can often be easier to provide examples of desired gestures than to describe embodied actions in program code. What should end-user designers learn about ML to effectively implement, debug, and refine new systems—and to understand what ML is capable of?

Finally, we must not forget that many stakeholders of ML systems are neither ML practitioners nor system builders. Indeed, the nature of models being built using millions or even billions of

peoples' data is that very few people are producing systems that potentially impact very many. This raises additional research questions relating to basic literacy for civic participation in a world where consequential decisions are being shaped by ML technologies. For example, *What should a judge know about ML to decide whether a sentencing recommendation system violates due process rights? What should a legislator know to improve laws regarding fairness in mortgage lending? What should an investor know before deciding to lend their money to a startup promising incredible innovations (e.g. self-driving cars)? What should an everyday citizen know to decide whether to allow a company to use its data in a particular way?*

ARTICLES IN THIS ISSUE

The three articles in this special issue address a breadth of topics.

The article by Sulmont, Patitsas, and Cooperstock investigates instructors' experiences of teaching ML to nonmajors. The authors interviewed instructors about their learning goals and experiences of what is easy and difficult to teach. Notably, they found that instructors were not very concerned about the difficulty of teaching ML algorithms; rather, they were more concerned with the difficulty of teaching students how to make good design decisions in the process of applying ML to real problems.

The article by Fiebrink argues for the importance of teaching ML to creative practitioners, including artists and musicians working with digital technology, then proposes a set of learning objectives for such students and a set of teaching strategies that the author has used in her own teaching. The article then draws on data collected from online and in-person courses to explore how the proposed curriculum supported student learning and to reveal information about what learners found challenging.

The article by Saltz et al. begins by examining how ethics are currently integrated into ML courses. It then proposes a framework for helping ML students identify ethical questions within ML projects, and it presents a pilot evaluation of this framework with 85 students. The article further describes three example ML course modules that educators can use to incorporate ethics topics into existing ML courses.

CONCLUSION

We have found that often when we tell people that we are teaching ML or are studying how people learn ML, we are understood to mean that we are helping people to learn about ML algorithms, including the theory and mathematics thereof. As the articles in this issue demonstrate, efforts to teach ML, and to study the learning of it, can be far broader than that, including a variety of topics related to apply and critically evaluate the application of ML to interesting problems.

This special issue is the first collection of research addressing this breadth of concerns for ML education. We sincerely hope it will not be the last. Simultaneously, we are optimistic that the breadth of ways in which ML can be applied, including within classical computing problems (e.g., to optimize processor microarchitectures) and in human-centered computing systems (e.g., to create customized prosthetic devices or to tune social media systems for maximum engagement, political consequences be damned) will increasingly necessitate desiloization of ML education. ML education should not only be a topic of study for computing education researchers, but one that engages math and statistics educators, political scientists, arts educators, linguists, and manifold others whose technical and humanistic perspectives can enable us to develop and share a more robust body of knowledge.

REFERENCES

Association for Computing Machinery and IEEE Computing Society. 2013. Computer Science Curricula 2013. Retrieved July 31, 2019 from https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf.

- F. Bernardo, M. Zbyszynski, R. Fiebrink, and M. Grierson. 2017. Interactive machine learning for end-user innovation. In *Proceedings of the AAAI Symposium Series: Designing the User Experience of Machine Learning Systems*.
- D. Bohn. 2019. Amazon says 100 million Alexa devices have been sold—What's next? *The Verge*. Retrieved July 31, 2019 from <https://www.theverge.com/2019/1/4/18168565/amazon-alexa-devices-how-many-sold-number-100-million-dave-limp>.
- J. Clement. 2000. Analysis of clinical interviews: Foundations and model viability. In *Handbook of Research Methodologies for Science and Mathematics Education*, R. Lesh and A. Kelly (Eds.). Erlbaum, Hillsdale, NJ, 341–385.
- B. Cowgill and C. E. Tucker. Forthcoming. Economics, fairness and algorithmic bias. *Journal of Economic Perspectives*.
- J. Dean, D. Patterson, and C. Young. 2018. A new golden age in computer architecture: Empowering the machine-learning revolution. *IEEE Micro* 38, 2 (2018), 21–29.
- John J. Dudley and Per Ola Kristensson. 2018. A review of user interface design for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems* 8, 2 (2018), Article 8.
- R. Fiebrink, P. R. Cook, and D. Trueman. 2011. Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 147–156.
- C. Goodwin. 1994. Professional vision. *American Anthropologist* 96, 3 (1994), 606–633.
- L. Haaranen and T. Lehtinen. 2015. Teaching git on the side: Version control system as a course platform. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. ACM, New York, NY, 87–92.
- E. Hutchins. 1995. *Cognition in the Wild* (No. 1995). MIT Press, Cambridge, MA.
- Indeed Editorial Team. 2019. Top 10 AI Jobs, Salaries and Cities. Retrieved July 31, 2019 from <http://blog.indeed.com/2019/06/28/top-10-ai-jobs-salaries-cities/>.
- Kurt Jacobson, Vidhya Murali, Edward Newett, Brian Whitman, and Romain Yon. 2016. Music personalization at Spotify. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, New York, NY, 373.
- D. A. Jiménez and C. Lin. 2002. Neural methods for dynamic branch prediction. *ACM Transactions on Computer Systems* 20, 4 (2002), 369–397.
- M. Kircher. 2016. Tinder will now show data on which of your pics get swipes. *New York Magazine*. Retrieved September 23, 2019 from <https://nymag.com/intelligencer/2016/10/tinder-smart-photos-show-which-pictures-get-swipes.html>.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1097–1105.
- C. O'Neil. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Broadway Books, Portland, OR.
- J. G. Politz, S. Krishnamurthi, and K. Fisler. 2014. In-flow peer-review of tests in test-first programming. In *Proceedings of the 10th Annual Conference on International Computing Education Research*. ACM, New York, NY, 11–18.
- R. S. Russ and T. O. B. Odden. 2017. Intertwining evidence- and model-based reasoning in physics sensemaking: An example from electrostatics. *Physical Review Physics Education Research* 13, 2, 020105-1–020105-14.
- S. Saria, A. Butte, and A. Sheikh. 2018. Better medicine through machine learning: What's real, and what's artificial? *PLoS Medicine* 15, 12, e1002721. <https://doi.org/10.1371/journal.pmed.1002721>.
- Amit Sharma and Baoshi Yan. 2013. Pairwise learning in recommendation: Experiments with community recommendation on LinkedIn. In *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, New York, NY, 193–200.
- B. Sherin. 2013. A computational study of commonsense science: An exploration in the automated analysis of clinical interview data. *Journal of the Learning Sciences* 22, 4 (2013), 600–638.
- P. Y. Simard, S. Amershi, D. M. Chickering, A. E. Pelton, S. Ghorashi, C. Meek, and J. Wernsing. 2017. Machine teaching: A new paradigm for building machine learning systems. arXiv:1707.06742.
- J. P. Smith III, A. A. Disessa, and J. Roschelle. 1994. Misconceptions reconceived: A constructivist analysis of knowledge in transition. *Journal of the Learning Sciences* 3, 2 (1994), 115–163.
- K. Starbird. 2019. Disinformation's spread: Bots, trolls and all of us. *Nature*. Retrieved July 31, 2019 from <https://www.nature.com/articles/d41586-019-02235-x>.
- R. Tromble and S. C. McGregor. 2019. You break it, you buy it: The naïveté of social engineering in tech—and how to fix it. *Political Communication* 36, 2, 324–332. DOI: [10.1080/10584609.2019.1609860](https://doi.org/10.1080/10584609.2019.1609860)
- C. Wills. 2019. Analysis of current and future computer science needs via advertised faculty searches for 2019. *Computing Research Association*. Retrieved July 31, 2019 from <https://cra.org/analysis-of-current-and-future-computer-science-needs-via-advertised-faculty-searches-for-2019/>.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv:1609.08144.
- A. Zimmermann-Niefeld, M. Turner, B. Murphy, S. K. Kane, and R. B. Shapiro. 2019. Youth learning machine learning through building models of athletic moves. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*. ACM, 121–132.