Distributed Training of Deep Learning Models

(Technical Paper)

Investigating the Politics of Deep Learning

(STS Paper)

A Thesis Prospectus Submitted to the

Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements of the Degree
Bachelor of Science, School of Engineering

Brian Yu
STS 4500, Spring 2020

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature _____ Date _____

Brian Yu

Approved _____ Date _____

Toluwalogo B. Odumosu, Department of Engineering and Society

I. Technical Project Description

Deep learning is currently one of the most active research areas in computing. Deep learning models are complex machine learning algorithms trained on very large datasets. These models are often trained on millions of data points by giving each data point and its label to the model, with the hope that the model will be able to generalize to predict the labels for data points it has never seen before. For example, AlexNet, which is often cited as the model that kicked off the deep learning revolution, is a model that labels the objects in an image. AlexNet was trained on ImageNet, a dataset comprised of 14 million images with manually annotated labels of the objects in the image. AlexNet was the first deep learning model able to outperform top non-deep learning methods, which it did by 10.8 percent (Krizhevsky et al., 2012).

Since the advent of AlexNet, researchers in academia and industry have been rushing to train deep learning models on larger and larger datasets and apply them to even more novel problems. The efficacy of deep learning algorithms increases with the amount of data that you can train the algorithm on and the complexity of the model. With larger datasets, deep learning models become more robust to outliers and overfitting, allowing them to generalize better to data that they have never seen before. More complex models with a higher number of parameters can learn more complex probability distributions of data and thus also generalize better. The tradeoff is that increasing model complexity and data set size can drastically increase the amount of time it takes to train deep learning models. Modern, industry-standard deep learning models can take weeks and even months of non-stop computation to fully train.

The field of distributed systems is another active research area in computing. Distributed systems are computer systems in which different components of the system run on different

computers that communicate over a network. They allow systems to scale beyond the computing capacity of a single computer and are what make modern large-scale computer systems possible. Google and Amazon are well-known examples of companies that heavily rely on distributed systems. Task scheduling is an area of study in which researchers build algorithms to more efficiently plan and place computation tasks on different computers in a distributed system. For example, given three hundred tasks with different hardware requirements and runtimes and a cluster of ten computers, an optimal scheduling algorithm would schedule the tasks on the computers so that all the tasks would finish in the shortest amount of time possible.

Given the high computational demands of deep learning models and the prevalence of distributed systems, one common approach to increasing the training speed of deep learning models is to leverage distributed systems in a process known as distributed training. There are many different types of deep learning models, each with its own unique set of computational demands. Thus, distributed training is a complex problem. Furthermore, given a cluster of computers and an arbitrary number of models, how can we train all of these models on the cluster so that they fully utilize cluster resources and finish training quickly? This is the main focus of my technical project. My project involves implementing state-of-the-art deep learning scheduling algorithms on both simulated and real computer clusters. These implementations provide a baseline to which researchers in my lab can compare their scheduling algorithms.

Deep learning tasks have unique computational requirements. Deep learning model training heavily utilizes graphics processing units (GPUs) and is cyclical in nature, having peaks and troughs of resource usage on timescales that are dependent on the specific model and batch of data that the model is training on. Furthermore, multiple models training on the same

computer may or may not interfere with each other's training speed, and training models on GPUs closer to the CPU may also increase performance (Xiao et al., 2018). Reconciling all of these factors that may affect training performance is the main challenge in creating distributed deep learning scheduling algorithms.

II. STS Research Question

Deep learning algorithms are being used in novel and powerful ways. They power the latest generation of autonomous vehicles, drive content recommendation algorithms that control what we watch and read online, and are used to analyze massive amounts of data in order to detect fraud and crime (Liu et al., 2017). Given their use on increasingly important problems, my STS research question asks the following: what are the political implications of these algorithms given their power over users and the political power of the tech companies that build and deploy them?

Deep learning models will have an increasing influence on our society as we become more reliant on them. They already heavily influence the content that we see online, via search and recommendation systems used by Google, Youtube, Twitter, and Facebook. How do these data-driven recommendation systems that aim to maximize user engagement based on prior data affect users? The STS field of user studies will be useful in examining the unique relationship that these deep learning models have with users, where users configure the algorithms and the algorithms configure the users.

These algorithms are designed to maximize user engagement. Oftentimes, this means surfacing content that agrees with a user's predisposed biases, even at the cost of correctness.

Winner's framework of the politics of artifacts will help analyze the politics and biases that these algorithms may exhibit.

Deep learning models are most effective when trained on large datasets using complex model architectures. Furthermore, there is a well-documented reproducibility crisis in the field of deep learning research (Barber, 2019). Thus, deep learning is uniquely suited for institutions that can collect massive datasets, afford large computer clusters, and hire researchers from top universities and companies who know the trade secrets of the field. STS studies of subpolitics will aid in investigating the potentially undemocratic nature of deep learning algorithms and the companies that create them.

III. STS Frameworks

User Studies

The field of user studies is concerned with the interactions between designers, technology, and users. One important concept is that of *scripts*, which are assumptions that designers make about users and the way that they use technology (Akrich, 1992). These assumptions create a framework of action that is prescribed for users. Sometimes these assumptions are incorrect and the technology is ill-designed for users. In other cases, technology can mold users and society in powerful ways.

In 1990, Steve Woolgar chronicled the production of a new computer system while acting as a project manager (Woolgar, 1990). Woolgar wrote that the "machine text is organized in such a way that 'its purpose' is available as a reading to the user." In other words, the machine is designed so that users can figure out how to use it as the designers intended. In the process of

creating a computer, the designers define their target users and establish suggestions and bounds of the users' actions. This is known as configuring the user.

The main challenge that faced the designers in both Woolgar's and Akrich's writing is that of knowing one's users. Designers go to great lengths to learn more about users lest their products fail to be adopted. Machine learning-driven recommendation algorithms are interesting under this framework because data is constantly being collected about users and used to improve the recommendations that they give. In turn, users are configured by those recommendations, forming a feedback loop. User studies will provide a framework with which to analyze these interactions.

Politics of Artifacts

In 1980, Langdon Winner wrote that technological artifacts can be political, even if they were not designed to be (Winner, 1986). He analyzed the case of a mechanical tomato harvester designed at Berkeley in the 1940s. While it produced a better yield than manual picking, it caused thousands of workers to lose their jobs. This outcome was not intended by the researchers, but reinforced the power of large growers and diminished that of the workers. Other technologies are inherently political. Some require a particular social structure in order to work and some are strongly aligned with a particular social system.

Given the potential of deep learning algorithms to automate work done by humans, it is obvious that these algorithms can be political, similar to the tomato harvester developed at Berkeley. Content recommendation algorithms may also exhibit political biases, even if unintended by their creators. Winner's framework will help analyze the politics of these algorithms and how they relate to society.

Subpolitics

The body of knowledge of science and technology is constantly growing and is showing no signs of slowing down or becoming any less important to society. As such, experts have increasing power to determine technological standards and regulations (De Vries, 2007). This may be problematic because these experts often are not elected public officials and may make these decisions behind closed doors. Thus, they set standards that deeply affect the public without involving the public in the decision-making process.

This rings true in the field of deep learning. In the absence of government regulation, tech companies are setting de facto industry standards in the use of deep learning algorithms. These companies may be over-optimistic about the abilities of their technology, leading to outcomes that would have been prevented by more stringent regulations, such as the fatal crash of a self-driving Tesla Model S in 2016 (Stilgoe, 2017). The STS field of subpolitics will aid in the examination of how these tech companies exert their influence on society and how the public, the government, and these subpolitical entities interact.

IV. Research Schedule

- **1/15/2021** - Finish reading about the history of AI
    - Relevant portions of The Quest for Artificial Intelligence: A History of Ideas and Achievements (2009) by Nilsson.
    - A Sociological Study of the Official History of the Perceptrons Controversy (1996) by Olazaran.
- **2/1/2021** - Review User Configuration

    - Configuring the User (1990) by Woolgar.
    - Engineering Knowledge: The Construction of Knowledge in Artificial Intelligence (1993) by Forsythe.
    - Of Robots and Humans: Creating User Representations in Practice (2020) by Fischer, Ostlund, and Peine.
- **2/15/2021** - Review Politics of Artifacts
    - Do Artifacts Have Politics? (1986) by Winner.
    - Research more empirical examples of AI and Politics.
    - Concerns about human agency, evolution, and survival (2018) by Anderson and Rainie.
    - The Ethical Character of Algorithms -- and What It Means for Fairness, the Character of Decision-Making and the Future of News (2018) by MacCarthy.
- **3/1/2021** - Research Big Tech Subpolitics
    - What Is Political In Sub-Politics? (2007) by De Vries.
    - Machine Learning, Social Learning, and the Governance of Self-Driving Cars (2017) by Stilgoe
    - Better, Nicer, Clearer, Fairer: A Critical Assessment of the Movement for Ethical Artificial Intelligence and Machine Learning (2019) by Greene, Hoffmann, and Stark

References

Akrich, M. (1992). The De-Scription of Technical Objects. In *Shaping technology, building society: studies in sociotechnical change*. Cambridge, Mass. u.a.: MIT Press.

Barber, G. (2019, September 16). Artificial Intelligence Confronts a 'Reproducibility' Crisis. Retrieved April 21, 2020, from https://www.wired.com/story/artificial-intelligence-confronts-reproducibility-crisis/

De Vries, G. (2007). What is Political in Sub-politics? *Social Studies of Science*, *37*(5), 781–809. doi: 10.1177/0306312706070749

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems*, *1*, 1097–1105. doi: 10.5555/2999134.2999257

Liu, W. E., Wang, Z. E., Liu, X. E., Zeng, N. E., Liu, Y. E., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, *234*, 11–26. doi: 10.1016/j.neucom.2016.12.038

Stilgoe, J. (2017). Machine Learning, Social Learning and the Governance of Self-Driving Cars. *Social Studies of Science*. doi: 10.1177/0306312717741687

Winner, L. (1986). *The whale and the reactor: a search for limits in an age of high technology*. Chicago: University of Chicago Press.

Woolgar, S. (1990). Configuring the User: The Case of Usability Trials. *The Sociological Review*, *38*, 58–99. doi: 10.1111/j.1467-954x.1990.tb03349.x

Xiao, W., Bhardwaj, R., Ramjee, R., Sivathanu, M., Kwatra, N., Han, Z., … Zhou, L. (2018). Gandiva: Introspective Cluster Scheduling for Deep Learning. *13th USENIX Symposium on Operating Systems Design and Implementation*.