Thanks to the rise and maturity of “Big Data” technology, the availability of big datasets on distributed computing systems attracts both academic researchers and industrial practitioners to apply more and more sophisticated machine learning techniques on those data to create higher value. Machine learning training, which summarizes the insights from big datasets as a mathematical model, is an essential step in any machine learning application. Due to the growing data size and model complexity, machine learning training demands increasingly high compute power and memory. In this dissertation, I present techniques that schedule computing tasks to better utilize network bandwidth, computation, and memory to improve training time and scale model size.

Machine learning training searches for optimal parameter values that maximize or minimize a certain objective function by repetitively processing the training dataset to refine these parameters in small steps. During this process, properly bounded error can be tolerated by performing extra search steps. Bounded error tolerance allows trading off learning progress for higher computation throughput, for example, by parallelizations that violate sequential semantics and such trade-offs should be made carefully. Widely used machine learning frameworks, such as TensorFlow, represent the complex computation of a search step as a dataflow graph to enable global optimizations, such as operator fusion, data layout transformation, and dead code elimination. This dissertation leverages bounded error tolerance and the intermediate representation of training computation, such as dataflow graphs to improve training efficiency.

First, I present a communication scheduling mechanism for data-parallel training that performs fine-grained communication and value-based prioritization for model parameters to reduce inconsistency in parameter values for faster convergence. Second, I present an automated computation scheduling mechanism that executes independent update computations in parallel with minimal programmer effort. Communication scheduling achieves faster convergence for data-parallel training and when applicable, computation scheduling achieves even faster convergence using much less network bandwidth. Third, I present a mechanism to schedule computation and memory allocation based on the training computation’s dataflow graph to reduce GPU memory consumption and enable training much larger models without additional hardware.

Thesis Committee:
Garth A. Gibson, Co-chair
Eric P. Xing, Co-chair
Phillip B. Gibbons
Gregory R. Ganger
Vijay Vasudevan, Google Brain