Machine Learning Parallelization Could Be Automated, Performant, and Easy-to-use

As models and data grow bigger, ML parallelization is more essential than ever. However, the amount of engineering effort and domain knowledge required for scaling up ML is often underestimated. The marginal cost for developing specialized systems with hand-tuned parallel strategies is extremely high in the face of emerging models and heterogeneous cluster setups.

In this talk, I will present a better way to build better ML systems. I view ML system building as an optimization over a parallel strategy space, with the objective of maximizing the system “goodput”, conditioned on model and cluster configurations. I show that by formulating each piece in the optimization as math representations, we can make it solvable using existing tools. Unlike specialized systems, this formulation enables building generic ML compilers that automate ML parallelization, generalize to many models, and achieve strong performance, simultaneously. In particular, I’ll describe two compiler systems: Alpa and Cavs, which automate model parallelism on large-scale distributed clusters, and the batching of dynamic neural network computation on accelerators, respectively. My open-source artifacts have been used by organizations such as AI2, Meta, and Google, and parts of my research have been commercialized at multiple start-ups including Petuum and AnyScale.

Bio: Hao Zhang is a postdoc researcher at UC Berkeley working with Ion Stoica. He completed his Ph.D. at CMU where he worked with Eric Xing. His research interests are in the intersection of machine learning and systems, with the focus on improving the performance and ease-of-use of today’s distributed ML systems. Hao’s research has been recognized with an NVIDIA pioneer research award at NeurIPS’17, and the Jay Lepreau best paper award at OSDI’21.

Host: Sanjay Krishnan