In recent years, the pace of innovations in the fields of machine learning has accelerated. To cope with the sheer computational complexity of training large ML models on large datasets, researchers in SysML have created algorithms and systems that parallelize ML training and inference over multiple CPUs or GPUs, or even multiple computing nodes over a network (distributed machine learning). As ML and deep learning models become more structurally complex, these systems have struggled to provide excellent all-round performance on a wide variety of models. In this thesis, we propose a simple design principle, adaptive parallelism, to guide the design of algorithms and systems for large-scale distributed ML. Following this principle, we derive a series of new parallelization strategies that interpolate existing ML parallelisms and adapt to the model, algorithm and cluster specifications. We examine these strategies and show that they boost the scalability and efficiency for one order of magnitude across a diverse set of models, environment, and distributed ML workloads, and open up the space for new model and algorithm design.

Generalized from multiple instantiations of this methodology, we develop ways of expressing various ML parallelisms, originated from different aspects, including synchronization architecture, model partitioning and placement, consistency, to name a few, within a unified representation. We demonstrate that the representation allows for rapid composing of adaptive parallelization strategies for unseen models from existing parallelisms, simplification of parallel ML programming, and leads to system implementations with better abstraction. Finally, we identify that ML scale-up is usually underestimated in terms of the amount of knowledge and time required to map between an appropriate distribution strategy to the model. To overcome this obstacle, we propose to investigate learning-based methods for distribution strategy search over the space spanned by representation, in order to automate ML parallelization.