Towards ML-Integration and Training Patterns for AI-Enabled Systems

Sven Peldszus[1][0000−0002−2604−0487], Henriette Knopp[1], Yorick Sens[1], and Thorsten Berger[1,2][0000−0002−3870−5167]

[1] Ruhr University Bochum, Germany `name.surname@rub.de`
[2] Chalmers | University of Gothenburg, Sweden

Abstract. Machine learning (ML) has improved dramatically over the last decade. ML components have become a fundamental part of intelligent software systems, many of which are safety-critical. Since ML components have complex lifecycles, they require dedicated methods and tools, such as pipeline automation or experiment management. Unfortunately, the current state of the art is *model-centric*, disregarding the challenges of engineering systems with multiple ML components that need to interact to realize complex functionality. Consider, for instance, robotics or autonomous driving systems, where perception architectures can easily incorporate more than 30 ML models. Developing such multi-ML systems requires architectures that can integrate and chain ML components. Maintaining and evolving them requires tackling the combinatorial explosion when re-training ML components, often exploring different (hyper-) parameters, features, training algorithms, or other ML artifacts. Addressing these problems requires *systems-centric* methods and tools. In this work, we discuss the characteristics of multi-ML systems and the challenges of engineering them. Inspired by such systems in the autonomous driving domain, our focus is on experiment-management tooling, which supports tracking and reasoning about the training process for ML models. Our analysis reveals their concepts, but also their limitations when engineering multi-ML systems, especially due to their model-centric focus. We will discuss possible ML training and integration patterns to facilitate the effective and efficient development, maintenance, and evolution of multi-ML systems. Furthermore, we will describe real-world multi-ML systems, providing early results from identifying and analyzing open-source systems from GitHub.

Keywords: ML-Enabled Systems · ML Asset Management · Maintenance · Evolution.