PyPads: Bootstrapping Community-Driven Open Science for Machine Learning

Thomas Weißgerber, Mehdi Ben Amor,
Christofer Fellicious, Michael Granitzer
{thomas.weissgerber, mehdi.benamor, christofer.fellicious,
michael.granitzer}@uni-passau.de

The field of machine learning progresses at an ever-accelerating pace. Despite algorithmic advancements, a need for improvement in the infrastructure supporting machine learning development and research becomes increasingly apparent. Nowadays, most machine learning experiments are done in an ad-hoc manner and results are mostly communicated through published literature, thereby providing only limited experimental details. At the same time, tracking and communicating an often inherently explorative scientific process is emerging to be a task with considerable effort. When solved insufficiently, understand-ability and reproducibility of experiments as well as the current state of research are hindered. As a remedy, the Open-Science movement encourages scientists to practice virtues like sharing Open-Notebooks. Nevertheless, essential features required to adhere to Open-Science principles are in practice often omitted due to several reasons, like e.g. time constraints, inexperience or low incentives in the community. Even when such features exist, artefacts are often logged in proprietary formats excluding provenance information and impeding automatic evaluation. Novel tools developed to tackle these challenges tend to be problem specific and employing them requires a significant time investment in their development and limits flexibility.

PyPads is an open source framework that provides an infrastructure to extend experimental setups with logging, communication and analysis features in a unobtrusive, community-driven way. PyPads builds on existing tools to offer automated, semantically enriched output logs of experiments. With MLFlow, PyPads employs a backend for the experiment life cycle, providing added benefits of automated, community-driven, and semantically structured logging, which enable complex queries. These extensions are implemented, while maintaining compatibility to already developed visualizations and comparison strategies, like the databricks jupyter notebook extension. To inject tracking capabilities, we employ 'YAML' based mapping files which are human readable code references combined with logging hooks and additional meta information about their contexts. Similar to classic XML mapping files in the software engineering domain, these files can be considered as a structural configuration including descriptive metadata. An ecosystem of customized mappings can be used to extend PyPads to different python-based frameworks. A goal of PyPads is to bootstrap the documentation, sharing, and discussion process for scientific experiments in general, while providing in depth features for the Machine Learning domain. With the provided infrastructure, we aim to support and check for crucial artefacts for Open-Access and Open-Data, identified in preceding works. These include checklists and properties of the experimental setups. Additionally, loggers give access to a schema of their output ensuring a structured model and querying capabilities to provide a critical contribution to the understand-ability

and access-ability of experiments. Ontology integration will ease the placement of the experiments' specifics and impacts in the domain, while the Open-Source aspect is furthered by incorporating source references and version information to PyPI or git repositories. In further steps compiling Open-Notebooks for the research process will be supported in form of jupyter notebook plugins and report generation. We hope this leads to an Open-Data pool helping users glean important insights from single experiments, relationships and the field as a whole.