GUILD AI: SIMPLE REPRODUCIBILITY IN MACHINE LEARNING

Garrett Smith [*1]

ABSTRACT

Guild AI is an open source tool for running, tracking, and comparing machine learning experiments. Guild helps researchers and engineers do their work more efficiently without imposing additional configuration and system requirements. By simplifying experiment tracking, Guild hopes to encourage more consistent reproducibility and improved collaboration in machine learning research.

1 GUILD AI

Experiments. Guild runs experiment trials as operating system processes and tracks results in separate run directories. Each run directory contains artifacts created during the trial along with metadata such as model information, start and stop times, logs, and run status. Run artifacts commonly include training checkpoints, prepared data sets, generated content such as images and audio, logs, and project source code snapshots.

Analysis. Guild provides facilities to analyze and compare experiments. Users can study experiment outputs such as training and validation loss, accuracy, required memory, batch latency as well logs and generated files. Guild also supports detailed comparison across experiment inputs including changes in model architecture, hyperparameters, data sets, and source code.

Sharing Results. Guild helps researchers share results in three ways ways:

- Project maintainers use *Guild files*—simple YAML files included with project source—to document and automate project capabilities such as supported models, operations, hyperparameter defaults and optimization search space.
- Project maintainers create Python *packages* that are uploaded to PyPI and installed by others for reproducing results.
- Users *publish experiment results* to secure, remote locations over SSH or cloud services like S3 where they are studied and compared by others.

Command Line Interface, Visualization, API. Guild's primary user interface is a command line interface, which integrates efficiently with other system tools. Guild additionally provides a graphical, web based dashboard to explore and analyze runs and integration with TensorBoard. Guild also exposes functionality in a Python API for advanced scripting and application developers, though this interface is not required for general use.

Cross Framework and Language Support. Guild supports any learning script that generates experiment results. Models may be implemented in any framework or library including but not limited to TensorFlow, PyTorch, scikit-learn, Keras, MXNet, or language such as Python, R, Go, and Java.

1.1 Reproducibility

Guild supports reproducibility by automating experiments, capturing results, and providing tools for analysis, comparison, and sharing. A typical workflow in Guild is:

1. Run trials for a baseline model
2. Run trials for novel work
3. Compare and analyze results
4. Optionally package and distribute project work to further simplify reproducibility

1.2 Users

Guild is used by researchers and engineers to run, track, and compare machine learning experiments. It is useful to anyone who needs to compare novel work to baseline results.

1.3 Related Work and Differences

There are several open source projects that address reproducibility in machine learning. Prominent projects include:

- ModelDB (Vartak, 2017)
- MLFlow (MLFlow, 2018)
- Polyaxon (Polyaxon, 2018)
- datmo (datmo, 2018)

[*]Equal contribution [1]Garrett Smith, Chicago, USA. Correspondence to: Garrett Smith <garrett@guild.ai>.

Proceedings of the 2^{nd} SysML Conference, Palo Alto, CA, USA, 2019.

Automation tools in machine learning commonly require tool-specific changes to project source code. This generally includes adoption of a Python API for controlling experiments, accessing hyperparameters and logging results. Automation tools may further require additional system software such as databases, container management systems, and job schedulers.

Each requirement that a tool imposes on a user presents a barrier to the goal of systematic machine learning. In some cases, users may view the cost of supporting reproducibility to outweigh the benefits.

To encourage systematic machine learning, Guild runs unmodified learning scripts without requiring the installation and maintenance of external systems. Once Guild is installed, for example, an unmodified Python script `train.py` that supports hyperparameters `n_layers` and `lr` can be run using:

```
$ guild run train.py n_layers=2 lr=1e-4
```

With this command, the user generates a unique experiment with the specified hyperparameters, which can be analyzed, compared to other trials, and shared.

Guild additionally supports easy packaging, distribution, and installation of research work in the support of machine learning reproducibility. To our knowledge, no other tool provides this feature.

2 DEMONSTRATION

We propose an interactive demonstration that highlights the benefits of fast, easy experiment reproduction. Throughout the session, the presenter refers to two personas—represented visually with placards—to help viewers maintain context during various scenario:

- *Arya*. A researcher who has published state-of-the-art results of a novel architecture on a standard benchmark data set.
- *Clegane*. A researcher working on a survey paper that includes Arya's work.

While demonstrating a scenario, the presenter highlights how Guild supports reproducibility in two areas:

- *Influence*. Work that is easier to reproduce is more likely to influence others.
- *Productivity*. Tools that support fast and easy reproducibility are applicable to research in general: more experiments generate more data, which researchers can use to advance their work.

Scenarios. The demonstration is structured flexibly to encourage questions, input, and redirection from the audience. While listed below sequentially, scenarios are presented and discussed in any order according to audience interest.

- *Arya* uses Guild to run training and validation experiments, collecting results—both positive and negative—and announces her findings in a celebrated invited talk at SysML. Her work is fully reproducible when she commits her code to GitHub.
- *Clegane* conducts a survey of related work, reproducing published results where possible. He clones Arya's repository and uses Guild to reproduce her findings. He further experiments with changes to Arya's model and hyperparameters to deepen his understanding for his report.
- *Clegane* contacts Arya to discuss her work and present questions he has based on his own experiments. He uses Guild to publish hist results to S3 where Arya can access them.
- *Arya* pulls Clegane's results from S3 and uses Guild to study the differences between their results and uses this information to answer his questions.
- *Clegane*, to promote his field of study and encourage further research, creates an installable package that others can use to reproduce Arya's work along with his own findings.

2.1 Equipment

The presenter supplies a laptop with an HDMI interface and brings supporting graphical placards and other visual aids to reference during scenarios. To avoid problems with network connectivity, the demonstration is disconnected from the network. Demonstrations of networking features (e.g. copying experiment results to and from S3) are run locally over a proxy, but are nonetheless live and unmocked.

REFERENCES

datmo. datmo, 2018. URL https://github.com/datmo/datmo.

MLFlow. Mlflow, 2018. URL https://github.com/mlflow/mlflow/.

Polyaxon. Polyaxon, 2018. URL https://github.com/polyaxon/polyaxon/.

Vartak, M. MODELDB: A system for machine learning model management. In *CIDR 2017, 8th Biennial Conference on Innovative Data Systems Research, Chaminade, CA, USA, January 8-11, 2017, Online Proceedings*, 2017.