Extension of a Machine Learning Experiment Management Tool

The Sacred Infrastructure for Computational Research

Martin Chovanec
Czech Technical University in Prague
chovamar@fit.cvut.cz

Supervisor: Klaus Greff
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)

Abstract

The aim of the project was to extend Sacred, an existing software tool for computational research that helps machine learning researchers with managing configuration, running, and maintaining results of their experiments. New functionality to Sacred has been added to provide its users with a standardized way of tracking experiment metrics and to simplify work with Sacred and the TensorFlow computational library. Furthermore, a web dashboard for Sacred has been created to enable browsing the collected results: the Sacredboard.

Introduction

A major part of machine learning research involves a number of computational experiments run with many different hyperparameter settings. Due to deadline pressure and the inherently unpredictable nature of research, there is usually little incentive for researchers to build robust infrastructures. As a result, research code often evolves quickly and compromises essential aspects like bookkeeping and reproducibility.

Sacred aims to fill the gap for tackling different aspects of the process by providing a central infrastructure for running computational experiments. Sacredboard is a user interface for browsing the Sacred data.

Main Objectives

1. Implementing a new Sacred API for collecting experiment monitoring information.
2. Simplification of using the tool with the TensorFlow library for computational intelligence.
3. Developing a dashboard to provide overview of running and past experiments and exploring their results

1 Sacred

Sacred is an open source Python framework to solve some of the frequent challenges of computational research. It is independent of the choice of machine learning libraries, aiming to offer maximum convenience while minimizing boilerplate code. Its configuration process allows integration with other tools, such as Labwatch for hyperparameter optimization. Through storage of run information in a central database, query and sorting functionality for bookkeeping becomes available. This further enables downstream analysis and allows other tools, such as Sacredboard, to provide a user interface for organizing results.

1.1 Configuration

An important goal of Sacred is to make it convenient to define, update and use hyperparameters, called the *configuration* of the experiment. The main way to set up the configuration is through decorated functions:

Listing 1: Configuration

```
@ex.config
def cfg():
  learning_rate = 0.1
  log_dir = 'log/NN{}'.format(learning_rate)

@ex.automain
def main(learning_rate, log_dir):
  ... # <= experiment code here
  return 42
```

This experiment is ready to be run and would return a *result* of 42. It already features an automatically generated command line interface, collects relevant information about dependencies and the host system, and can do bookkeeping.

1.2 Bookkeeping

Sacred accomplishes bookkeeping through attaching observers of the experiment. Partial data is captured even in the case of failures.

Sacred ships with observers for MongoDB, SQL database, or locally file storage. The interface is, however, extensible.

2 Sacredboard

Sacredboard provides a convenient way for browsing runs of experiments stored in a Sacred MongoDB database. It consists of a lightweight Flask-based web server that can be run on any machine with access to the database. The hosted web-interface shows a table view of both running and finished experiments, which are automatically updated. Sacredboard shows the current state and results, and offers a detail view that includes configuration, host information, and standard output of each run.

2.1 Filtering

Experiments can be filtered by status to, for example, quickly remove failed experiments from the overview. Sacredboard also supports filtering by config values, in which case the user specifies a set of properties names and conditions.

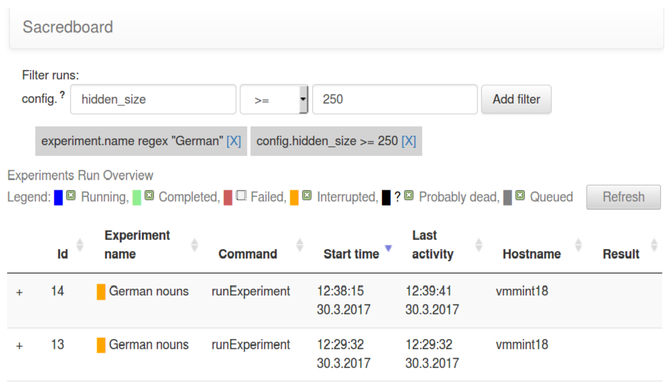

Figure 1: Sacredboard Interface

2.2 The Details View

Clicking on any of the displayed runs expands the row to a details-view that shows the hyperparameters used, information about the machine, the environment where the experiment was run, and the standard output produced by the experiment. The view is organised as a collapsible table, allowing dictionaries and arrays to be easily browsed. If TensorFlow tracking was enabled during the run, t is possible to launch TensorBoard directly from the Run detail view.

Sacredboard can also visualize metrics such as accuracy or loss if they are tracked using the Sacred Metrics API.

Listing 2: Capturing Metrics Data in Sacred Metrics API

```
_run.log_scalar("test.accuracy", 35.25, step=50)
```

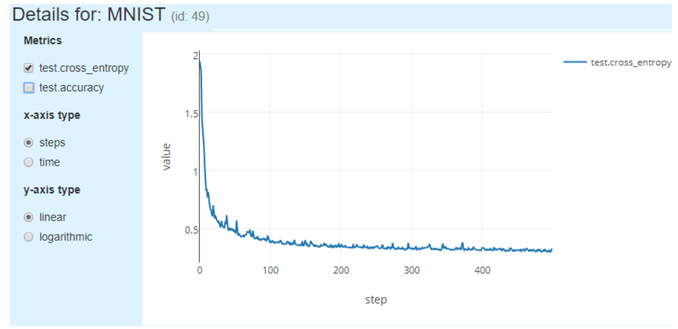

Figure 2: Sacredboard Metrics Plot

Conclusion

Sacred is an open source Python framework which aims to provide infrastructure for computational experiments with minimal boilerplate code and maximum convenience. It was, however, missing a proper user interface and its users had to rely on their own database queries to access records of the experiments. To solve the issue, the Sacredboard web dashboard has been created as a part of the thesis and Sacred itself has been extended to track new types of information.

Possible improvements include support for more complex experimental setups, like having separate training and evaluation scripts as is common with large TensorFlow models.

Sacredboard should be extended to fully support other Sacred backends, to better handle automatic updates and ideally, to integrate with the TensorFlow library in a way that does not depend on TensorFlow internals.

References

[1] Google Inc. Tensorflow. [online]. [Online; accessed 06-June-2017].

[2] Klaus Greff, Aaron Klein, Martin Chovanec, Frank Hutter, and Jürgen Schmidhuber. The Sacred Infrastructure for Computational Research. In Katy Huff, David Lippa, Dillon Niederhut, and M Pacer, editors, *Proceedings of the 15th Python in Science Conference*, pages 49 – 56, 2017.

[3] MongoDB, Inc. Mongodb, 2017. [Online; accessed 06-June-2017].

Acknowledgements

Access to computing and storage facilities owned by parties and projects contributing to the Czech National Grid Infrastructure MetaCentrum provided under the programme Projects of Large Research, Development, and Innovations Infrastructures (CESNET LM2015042) is greatly appreciated.

The text of the poster is mostly based on our paper. [2]

The work of authors of Sacred has partly been supported by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme under grant no. 716721, by the Euro- pean Commission under grant no. H2020-ICT-645403-ROBDREAM, and by the German Research Foundation (DFG) under Priority Programme Autonomous Learning (SPP 1527, grant HU 1900/3-1). Their research was supported by the EU project INPUT (H2020-ICT-2015 grant no. 687795).