Static Analysis of Shape in TensorFlow Programs (Artifact)

Sifis Lagouvardos
University of Athens, Greece
sifis.lag@di.uoa.gr

Julian Dolby
IBM Research, Yorktown Heights, NY, USA
dolby@us.ibm.com

Neville Grech
University of Athens, Greece
me@nevillegrech.com

Anastasios Antoniadis
University of Athens, Greece
anantoni@di.uoa.gr

Yannis Smaragdakis
University of Athens, Greece
smaragd@di.uoa.gr

Abstract

These instructions are intended for using the artifact for our ECOOP'20 paper entitled "Static Analysis of Shape in TensorFlow Programs". They can be used to run Pythia – the tool implementing the paper's analysis – on the paper's evaluation set demonstrating bug detection in the most precise configuration of our analysis as well as the precision of the analysis under different configurations.

2012 ACM Subject Classification Theory of computation → Program analysis; Software and its engineering → Compilers; Software and its engineering → General programming languages

Keywords and phrases Python, TensorFlow, static analysis, Doop, Wala

Digital Object Identifier 10.4230/DARTS.6.2.6

Funding We gratefully acknowledge funding by the European Research Council, grant 790340 (PARSE), and by the Hellenic Foundation for Research and Innovation (project DEAN-BLOCK).

Related Article Sifis Lagouvardos, Julian Dolby, Neville Grech, Anastasios Antoniadis, and Yannis Smaragdakis, "Static Analysis of Shape in TensorFlow Programs", in 34th European Conference on Object-Oriented Programming (ECOOP 2020), LIPIcs, Vol. 166, pp. 15:1–15:29, 2020.
`https://doi.org/10.4230/LIPIcs.ECOOP.2020.15`

Related Conference 34th European Conference on Object-Oriented Programming (ECOOP 2020), November 15–17, 2020, Berlin, Germany (Virtual Conference)

1 Scope

The artifact captures code that is also available in a public repository (`https://bitbucket.org/yanniss/doop/`). The latest version can be obtained online. The artifact snapshot the state of the code at the time of paper publication and can be used to reproduce two experiments from our paper; with the first one demonstrating precise bug detection and the second one showcasing the effect different analysis configurations have on precision.

1.1 Bug Detection

The first experiment includes analyzing our evaluation set using the most precise configuration of our analysis (1-call-site-sensitive with a context-sensitive heap and full-tensor-precision).

Table 2 in our paper summarizes the results for the 14 input programs in our evaluation set.

1.2 Precision

The second experiment demonstrates the effect different analysis configurations have in the precision of the analysis, reproducing the experiment shown in section 7.3 (and Figure 2) of our paper. We partly reproduce the results of the chart in Figure 2. The reason we cannot fully reproduce these results is that the false positives of our analysis require manual labor to be identified. For this reason we report the imprecision metrics that can be automatically produced by our analysis.

2 Content

The artifact contains the Doop program analysis framework that includes Pythia, the dataset of "An empirical study on TensorFlow program bugs"[1] (part of which is our evaluation set), the full artifact documentation in markdown and PDF formats, and scripts for running the analyses and post-processing their results.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS).

4 Tested platforms

Our artifact is bundled for AMD64 Linux using an Ubuntu 18.04 Docker image. If you do not already have Docker installed, follow the official installation instructions for your Linux system (can be found at `https://docs.docker.com/install/`). We've successfully tested our artifact using machines that have 16 GBs of RAM or more. Has been tested using ubuntu 18.04 and Windows 10 using WSL and WSL2.

5 License

6 MD5 sum of the artifact

12c251ecc51a5c6dc8bf1fe7ff2e873a

7 Size of the artifact

1.3 GiB

References

1 Yuhao Zhang, Yifan Chen, Shing-Chi Cheung, Yingfei Xiong, and Lu Zhang. An empirical study on tensorflow program bugs. In *Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2018, page 129–140, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3213846.3213866.