CS 846 Software Engineering for Big Data and AI

Toward Understanding Deep Learning Framework Bugs

Jun Lim

20870249

Deep learning (DL) frameworks are the foundation for building DL programs and models. Bugs in these critical frameworks can cause unexpected behaviors in any DL program or model that relies on them, and it is crucial to understand these bugs to assure the quality of DL frameworks. Several studies have been done on investigating DL program bugs, but little has been done on DL framework bugs.

This paper aims to conduct a comprehensive study to analyze and understand DL framework bugs. The authors collected 1000 bugs across four different frameworks (TensorFlow, PyTorch, MXNet, and DL4J) and deconstructed the DL framework into 5 levels and classified the findings into 13 root causes and 6 symptoms of DL-framework bugs:

- **Root Causes**: (1) type issue, (2) tensor shape misalignment, (3) incorrect algorithm implementation, (4) environment incompatibility, (5) API incompatibility, (6) API misuse, (7) incorrect assignment, (8) incorrect exception handling, (9) misconfiguration, (10) numerical issue, (11) concurrency issue, (12) dependent module issue, and (13) others
- **Symptoms**: (1) crash, (2) incorrect functionality, (3) build failure, (4) poor performance, (5) hang, and (6) unreported

Through the analysis, the authors obtained findings not just on the root causes and symptoms but also on the distribution and relationship between the causes and symptoms, which levels of DL frameworks are more prone to bugs, and whether bugs in different DL frameworks have common characteristics. From this, they obtained 12 major findings that contributed to a comprehensive understanding of DL framework bugs and the current status of DL framework testing practices, leading to a series of guidelines for better bug detection and debugging in DL frameworks.

To extend the usefulness of the findings, the authors have also designed and developed a prototype DL-framework testing tool called TenFuzz. In the preliminary evaluation on TensorFlow, TenFuzz was able to successfully detect six bugs, three of which were previously undetected, which demonstrates the practical relevance and potential impact of the findings.

In conclusion, the paper provided a comprehensive understanding of the complexities of bugs in DL frameworks. Through these empirically-backed studies, it provided a good foundation for future research into the detection and debugging of DL framework bugs.

Paper Commentary

The authors did the research in a systematic methodology, which involves detailed classification systems for bugs' root causes and symptoms, demonstrating rigorous research and giving the paper strong credibility. Furthermore, by studying multiple frameworks with diverse characteristics, the findings are made more generalizable across various DL frameworks. This diversity in analysis helps strengthen the paper's relevance to a broader audience and a range of applications. Thirdly, although the paper was lengthy (about 31 pages long), it provided a clear analysis and explanation for each finding, setting the stage for future research in the area.

On the flip side, the paper also has some weaknesses. The authors did the classification in a manual way, which may raise the idea of potential (human) bias in the categorization process. An approach to improving this would be to add an additional step that cross-verifies the classification to enhance the reliability of the results. Also, the proposed testing tool of TenFuzz was only evaluated against TensorFlow; it is unsure how reliable this tool is on other frameworks. Additional testing on other frameworks could help strengthen the reliability of the tool.

Overall, it was a great paper.