Towards Testing of Deep Learning Systems

Jianjun Zhao
Kyushu University
zhao@ait.kyushu-u.ac.jp

***Abstract*—Deep learning has achieved great success in many application domains such as image processing, speech recognition, and autonomous vehicles. However, how to ensure the reliability of deep learning systems remains an open problem. In this keynote, I introduce several automated testing techniques to ensure the reliability of deep learning systems.**

***Index Terms*—Deep learning, software testing, system reliability**

I. Introduction

Deep learning (DL) has achieved great success in many application domains such as image processing, speech recognition, and autonomous vehicles. However, how to ensure the reliability and security of deep learning systems remains an open problem. For example, an attacker could add adversarial perturbations often imperceptible to human eyes to an image to cause a deep neural network (DNN) to misclassify perturbed images. Traditional software represents its logic as control flows crafted by human knowledge, while a DNN characterizes its behaviors by the weights of neuron edges and the non-linear activation functions (determined by the training data). Therefore, detecting erroneous behaviors in DNNs is different from those of traditional software in nature, which necessitates effective analysis, testing and verification approaches [1]. We plan to take a multi-pronged approach to explore a deeper understanding of defects and adversarial examples in a DL system and propose some methods to guarantee its reliability and security. We next briefly introduce several automated testing techniques to ensure the reliability of DL systems.

Test Coverage Criteria for DL Systems. Currently, the testing adequacy of a DL system is usually measured by the accuracy of test data. Considering the limitation of accessible high-quality test data, good accuracy performance on test data can hardly provide confidence to the testing adequacy and generality of DL systems. Unlike traditional software systems that have clear and controllable logic and functionality, the lack of interpretability in a DL system makes system analysis and defect detection difficult, which could potentially hinder its real-world deployment. To this end, we propose *DeepGauge* [2], a set of multi-granularity testing criteria for DL systems, which aims at rendering a multi-faceted portrayal of the testbed. The in-depth evaluation of our proposed testing criteria is demonstrated on two well-known datasets, five DL systems, and with four state-of-the-art adversarial attack techniques against DL. The potential usefulness of *DeepGauge* sheds light on the construction of more generic and robust DL systems.

Test Data Generation of DL Systems. Similar to traditional software, DNNs could also exhibit incorrect behaviors caused by hidden defects causing severe accidents and losses. We propose *DeepHunter* [3], an automated fuzz testing framework for hunting potential defects of general-purpose DNN. *DeepHunter* performs metamorphic mutation to generate new semantically preserved tests, and leverages multiple pluggable coverage criteria as feedback to guide the test generation from different perspectives. To be scalable towards practical-sized DNNs, *DeepHunter* maintains multiple tests in a batch, and prioritizes the tests selection based on active feedback. The large-scale experiments demonstrate that *DeepHunter* can significantly boost the coverage with guidance, and generate useful tests to detect erroneous behaviors and facilitate the DNN model quality evaluation.

Test Data Quality of DL Systems. The standard way of evaluating DL models is to examine their performance on a test dataset. The quality of the test dataset is of great importance to gain the confidence of the trained models. Using inadequate test dataset, DL models that have achieved high test accuracy may still suffer from vulnerability against (adversarial) attacks. Mutation testing is a well-established technique to evaluate the quality of test suites in traditional software testing. We propose *DeepMutation* [4], a mutation testing framework specialized for DL systems. *DeepMutation* supports both source-level mutation testing, which is to slightly modify source (i.e., training data and training programs) of DL software, and also model-level mutation testing, which is to directly mutate on DL models without a training process.

Acknowledgment

This work was partially supported by 973 Program in China (No. 2015CB352203) and JSPS KAKENHI Grant 18H04097.

References

[1] K. Pei, Y. Cao, J. Yang, and S. Jana, “Deepxplore: Automated whitebox testing of deep learning systems,” in *Proc. 26th Symposium on Operating Systems Principles*, 2017, pp. 1–18.

[2] L. Ma, F. Juefei-Xu, F. Zhang, J. Sun, C. Chen, T. Su, M. Xue, B. Li, L. Li, Y. Liu, J. Zhao, and Y. Wang, “Deepgauge: Comprehensive and multi-granularity testing criteria for gauging the robustness of deep learning systems,” in *Proc. 33th IEEE/ACM Conference on Automated Software Engineering (ASE 2018)*, 2018, pp. 120–131.

[3] X. Xie, L. Ma, F. Juefei-Xu, H. Chen, M. Xue, B. Li, Y. Liu, J. Zhao, J. Yin, and S. See, “Deephunter: Hunting deep neural network defects via coverage-guided fuzzing,” in *arXiv:1809.01266*, 2018.

[4] L. Ma, F. Zhang, J. Sun, M. Xue, B. Li, F. Juefei-Xu, C. Xie, L. Li, Y. Liu, J. Zhao, and Y. Wang, “Deepmutation: Mutation testing of deep learning systems,” in *Proc. 29th IEEE International Symposium on Software Reliability Engineering (ISSRE 2018)*, 2018, pp. 120–131.