

1. DeepMutation: mutation testing of deep learning systems

Lei Ma (1); Fuyuan Zhang (2); Jiyuan Sun (3); Minhui Xue (2); Bo Li (4); Juefei-Xu, F. (5); Chao Xie (3); Li Li (6); Yang Liu (2); Jianjun Zhao (3); Yadong Wang (1)

Source: 2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE). Proceedings, p 100-11, 2018; **ISBN-13:** 978-1-5386-8321-7; **DOI:** 10.1109/ISSRE.2018.00021; **Conference:** 2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE), 15-18 Oct. 2018, Memphis, TN, USA;

Publisher: IEEE Computer Society, Los Alamitos, CA, USA

Author affiliation: (1) Harbin Institute of Technology, China (2) Nanyang Technological University, Singapore (3) Kyushu University, Japan (4) University of Illinois at Urbana-Champaign, Urbana, IL, United States (5) Carnegie Mellon University, Pittsburgh, PA, United States (6) Monash University, Melbourne, VIC, Australia

Abstract: Deep learning (DL) defines a new data-driven programming paradigm where the internal system logic is largely shaped by the training data. 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 confidence of the trained models. Using an inadequate test dataset, DL models that have achieved high test accuracy may still lack generality and robustness. In traditional software testing, mutation testing is a well-established technique for quality evaluation of test suites, which analyzes to what extent a test suite detects the injected faults. However, due to the fundamental difference between traditional software and deep learning-based software, traditional mutation testing techniques cannot be directly applied to DL systems. In this paper, we propose a mutation testing framework specialized for DL systems to measure the quality of test data. To do this, by sharing the same spirit of mutation testing in traditional software, we first define a set of source-level mutation operators to inject faults to the source of DL (i.e., training data and training programs). Then we design a set of model-level mutation operators that directly inject faults into DL models without a training process. Eventually, the quality of test data could be evaluated from the analysis on to what extent the injected faults could be detected. The usefulness of the proposed mutation testing techniques is demonstrated on two public datasets, namely MNIST and CIFAR-10, with three DL models. (0 refs)

Inspec controlled terms: learning (artificial intelligence) - program testing - software fault tolerance

Uncontrolled terms: deep learning systems - data-driven programming paradigm - internal system logic - training data - DL models - quality evaluation - DL systems - mutation testing framework - source-level mutation operators - model-level mutation operators - software testing - faults detection - DeepMutation

Classification Code: C6170K Knowledge engineering techniques - C6110B Software engineering techniques - C6150G Diagnostic, testing, debugging and evaluating systems

IPC Code: G06F9/44 - G06F11/36 - G06F15/18 - G06N5/04

Treatment: Bibliography (BIB) - Practical (PRA)

Database: Inspec

Data Provider: Engineering Village

Copyright 2018, The Institution of Engineering and Technology