Gregor Engels, Regina Hebig, Matthias Tichy (Hrsg.): SE 2023,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2023 131

DeepHyperion: Exploring the Feature Space of Deep Learning-based Systems through Illumination Search

Tahereh Zohdinasab[1] Vincenzo Riccio[2] Alessio Gambi[3] Paolo Tonella[4]

Abstract: In this extended abstract, we summarize our contributions to automated testing of Deep Learning-based systems published at the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA) in 2021 [Zo21a] and *just accepted* by the ACM Transactions on Software Engineering and Methodology (TOSEM) in 2022 [Zo22].

Deep Learning-based systems (DL Systems) find applications in safety-critical application domains and thus must be thoroughly tested. Existing DL system testing approaches can generate complex and fault-finding inputs but do not characterize them in a way that enables human interpretation and do not always consider test diversity. Our work addresses these challenges and can find effective and diverse test cases.

Keywords: Software testing; deep learning; search-based software engineering; self-driving cars

1 Methodology

Automatically and effectively testing DL systems, like self-driving cars, requires efficiently generating complex inputs, such as driving scenarios, and a strategy to balance exploration to discover untested behaviors and exploitation to identify highly effective test cases. Additionally, testing should produce results that developers can easily interpret and use to improve their design, for instance, by identifying features that are under-represented in training sets. We address the above challenges through DeepHyperion that leverages (1) model-based input representation for generating complex inputs, (2) Illumination Search to generate effective test cases while exploring the test space at large, and (3) metrics to quantify test input as well as DL systems under test's behavior features.

We proposed a two-step methodology that developers can follow to identify representative, discriminative, and quantifiable features that characterize the tests in an intuitive way. Our methodology consists of (i) Open Coding for selecting the independent variables describing the tests and (ii) Metric Identification for devising automatic procedures to quantify them. Having a way to quantify relevant test features, DeepHyperion implements the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites), the Illumination Search algorithm

[1] Università della Svizzera Italiana, Switzerland, tahereh.zohdinasab@usi.ch
[2] Università della Svizzera Italiana, Switzerland, vincenzo.riccio@usi.ch
[3] IMC University of Applied Sciences Krems, Austria, alessio.gambi@fh-krems.ac.at
[4] Universita della Svizzera Italiana, Switzerland, paolo.tonella@usi.ch

proposed by Mouret and Clune [MC15], to mutate a population of tests, called *seeds*. DeepHyperion aims to generate tests that maximize the likelihood of observing misbehaviors for each feature combination and produces N-dimensional maps that visualize the distribution of the fittest tests across the feature space. Those maps, in turn, enable developers to identify clusters of fault-revealing tests with similar features. Since DeepHyperion selects the individuals to mutate randomly, which is unbiased but ineffective, we extended it with a ranked selection scheme based on Contribution Score (CS), a novel metric that promotes individuals that contributed more to exploring the feature space.

2 Results

We evaluated DeepHyperion's effectiveness and efficiency in two application domains: recognition of hand-written digits, which is a classification problem, and steering angle prediction for self-driving cars in driving simulations, which is a regression problem. Our empirical results show that DeepHyperion over-performed existing approaches in both application domains by exposing more unique misbehaviors and exploring larger portions of the feature space. Additionally, our experiments confirmed that selecting individuals according to their Contribution Scores significantly improves DeepHyperion's efficiency. Regarding the usefulness of the N-dimensional feature maps generated by DeepHyperion, our evaluation shows how those maps help DL system developers in identifying shortcomings of training datasets and providing new data to expand them.

3 Data Availability

Our replication package [Zo21b] includes the original code we used for running the experiments and the data we collected during the evaluation of DeepHyperion. The latest version of the code, instead, is available on GitHub at:

`https://github.com/testingautomated-usi/DeepHyperion`

Bibliography

[MC15] Mouret, Jean-Baptiste; Clune, Jeff: Illuminating search spaces by mapping elites. CoRR, abs/1504.04909, 2015.

[Zo21a] Zohdinasab, Tahereh; Riccio, Vincenzo; Gambi, Alessio; Tonella, Paolo: DeepHyperion: exploring the feature space of deep learning-based systems through illumination search. In: Proc. of the Intl. Symposium on Software Testing and Analysis. ACM, pp. 79–90, 2021.

[Zo21b] Zohdinasab, Tahereh; Riccio, Vincenzo; Gambi, Alessio; Tonella, Paolo: , DeepHyperion Replication Package. URL: `https://doi.org/10.5281/zenodo.4742119`, May 2021.

[Zo22] Zohdinasab, Tahereh; Riccio, Vincenzo; Gambi, Alessio; Tonella, Paolo: Efficient and Effective Feature Space Exploration for Testing Deep Learning Systems. ACM Trans. Softw. Eng. Methodol., jun 2022. Just Accepted.