Uncertainty-Guided Testing and Robustness Enhancement for Deep Learning Systems

Xiyue Zhang
Peking University, China
zhangxiyue@pku.edu.cn

ABSTRACT

Deep learning (DL) systems, though being widely used, still suffer from quality and reliability issues. Researchers have put many efforts to investigate these issues. One promising direction is to leverage uncertainty, an intrinsic characteristic of DL systems when making decisions, to better understand their erroneous behavior. DL system testing is an effective method to reveal potential defects before the deployment into safety- and security-critical applications. Various techniques and criteria have been designed to generate defect-triggers, i.e. adversarial examples (AEs). However, whether these test inputs could achieve a full spectrum examination of DL systems remains unknown and there still lacks understanding of the relation between AEs and DL uncertainty. In this work, we first conduct an empirical study to uncover the characteristics of AEs from the perspective of uncertainty. Then, we propose a novel approach to generate inputs that are missed by existing techniques. Further, we investigate the usefulness and effectiveness of the data for DL robustness enhancement.

ACM Reference Format:
Xiyue Zhang. 2020. Uncertainty-Guided Testing and Robustness Enhancement for Deep Learning Systems. In *42nd International Conference on Software Engineering Companion (ICSE '20 Companion), May 23–29, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3377812.3382160

1 INTRODUCTION

In recent years, deep learning (DL) systems have achieved great success in a variety of domains. However, like traditional software systems, they still suffer from quality and reliability issues, which could lead to catastrophic consequences. Therefore, it is quite critical to reveal potential defects and further enhance the robustness of DL systems. In practice, testing is one of the effective techniques to reveal potential problems that exist in software systems. However, traditional guidance and techniques cannot be directly applied to such data-driven systems. To address this challenge, researchers have proposed several criteria to guide the generation of test inputs [11, 16, 20] and designed various methods to generate adversarial inputs that could trigger the defects hidden in DL systems [1, 5, 13, 17, 18, 24, 27].

ICSE '20 Companion, May 23–29, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7122-3/20/05…$15.00
https://doi.org/10.1145/3377812.3382160

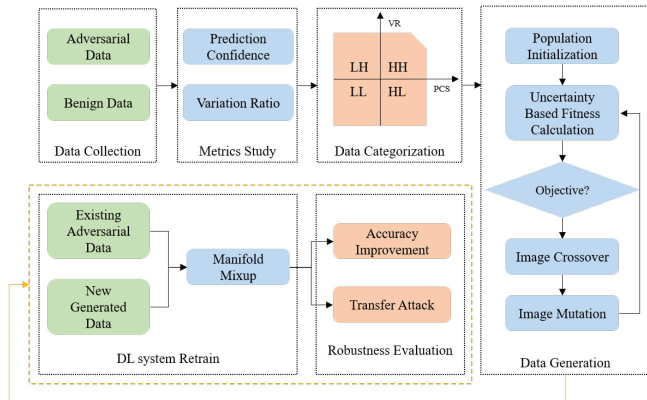

Figure 1: Workflow overview

Thus far, adversarial inputs obtained with small perturbations of the original data [5] are the most notable input cases that lead to DL decision errors. The understanding and explanation of such defect-triggers are still at an early age. Uncertainty [2], an intrinsic nature of DL, measures how confident the system is when making decisions over data inputs. It offers a new perspective for data characterization, and promisingly can be used to unveil the incurrence of decision errors in DL systems. However, the uncertainty of DL systems is still not well-studied and fails to reach their full potential in defect revealing and quality assurance.

To bridge this gap, (1) we first conduct an empirical study to understand the capability of different uncertainty metrics to characterize data inputs of DL systems, and formulate four types of uncertainty patterns, where existing inputs largely fall into two patterns. (2) Then, we leverage the uncertainty metrics as guidance to generate inputs, especially those with uncertainty patterns which are missed by existing techniques. (3) Finally, we investigate the effectiveness of the inputs with different uncertainty patterns in improving the accuracy and the robustness against transfer attacks. An overview is shown in Fig. 1. Note that the first two modules are completed and published in [30]. The follow-up work towards robustness enhancement is marked with yellow dashed lines.

2 APPROACH AND EVALUATION

We now introduce the experiment setup and present the approach for each workflow module.

2.1 Experiment Setup

We use three popular datasets (MNIST [14], CIFAR-10 [12], ImageNet [22]) and four DL systems (*LeNet-5* [14], *NIN* [15], *ResNet-20* [8], *MobileNet* [9]) as the studied objects. The test sets of MNIST, CIFAR-10, ImageNet naturally form the sets of benign examples (BEs) used in the data characteristic study. Four attack methods (FGSM [5], BIM [13], Deepfool [17], C&W [1]) and two testing techniques (TensorFuzz [18] and DeepHunter [27]) are utilized to generate adversarial inputs. Data generated by these two threads of techniques comprise the set of adversarial examples (AEs) used in the study.

2.2 Empirical Study of Data Characteristics

We first introduce the studied uncertainty metrics, followed by the uncertainty pattern categorization of the data prepared in 2.1.

Uncertainty metrics. Uncertainty captures more information possessed in the DL systems than merely a classification result. It reflects to what extent the model is uncertain about the decision against the input, which could be leveraged to characterize the behavior of data inputs. We collect and study four state-of-the-art uncertainty metrics from [2, 3, 10, 28]: *prediction confidence score* (PCS), *variation ratio* (VR), *predictive entropy* (PE), and *mutual information* (MI). For example, VR captures the dispersion from a specified label based on multiple system executions. Given a DL system D, an execution number T, a specified label l, and an input x, $VR(x, D) = 1 - \frac{\sum_t \mathbb{1}[L_{D^t}(x)=l]}{T}$, where L_{D^t} denotes the $t\text{-}th$ prediction result by D and $\mathbb{1}[]$ is the indicator function.

Uncertainty pattern of existing data. PCS and VR stand out from the collected uncertainty proxies in achieving better differentiating performance on AEs/BEs. It then leads to a data categorization method from two angles: PCS captures the prediction confidence in terms of single-shot DL system execution; and VR captures the Bayesian uncertainty based on the statistical multi-shot executions. These two metrics are then compositely used to categorize data into four patterns: low PCS / high VR (LH), high PCS / high VR (HH), high PCS / low VR (HL), and low PCS / low VR (LL).

We categorize the collected data to understand the characteristics of the generated data by existing techniques. We find that BEs mostly fall into HL pattern, while AEs mostly fall into LH pattern. Compared with the AEs generated through attack methods, those by testing techniques could trigger more diverse uncertainty behavior. The evaluation results demonstrate the necessity and usefulness of testing methods for DL systems.

2.3 Uncertainty-Guided Test Generation

The categorization results of the existing data lead to the following questions: (1) whether data inputs of the other uncertainty patterns could be generated; (2) whether data with such uncommon patterns could penetrate DL defense mechanisms more effectively, thus uncovering deeper hidden defects.

To address the first question, we adopt the Genetic Algorithm (GA) to generate the uncovered data inputs with uncertainty metrics as guidance. The key procedures are summarized here: (1) *Population construction.* We initialize the population by randomly adding noise to original input seeds, meanwhile using L_∞ norm to constrain the perturbation to the seed images. (2) *Fitness calculation.* For each uncovered pattern, we design a piece-wise fitness function to fulfill the optimization objective. (3) *Selection and crossover.* We use the tournament selection strategy to select candidate samples with the best fitness value, on which the crossover is then performed by randomly exchanging the corresponding pixels. (4) *Termination.* The test generation process terminates until the objective is satisfied or the given computation resources exhaust.

To address the second question, we perform comparison experiments on the newly generated data and existing data towards attacking DL systems equipped with a set of defense techniques [4, 6, 7, 19, 21, 26, 29]. Compared with data of common patterns, the uncommon data generated by the proposed approach could reveal 35% more hidden defects on average and up to 79% in certain scenarios.

2.4 Robustness Enhancement

Natural follow-up research questions are 1) whether the generated data could be rendered to enhance the robustness of DL systems more effectively, and 2) whether data with different uncertainty patterns differentiate from each other in terms of robustness enhancement capabilities.

Robustness criteria. The robustness enhancement is evaluated by measuring two criteria: 1) the accuracy improvement of retrained DL systems on the test sets; 2) the calibration rate of transfer attacks, i.e. the success rate of classification on transfer attack data [23].

DL system retrain. To address the first question, we carry out a comparison experiment towards robustness enhancement effectiveness between the newly generated data and existing adversarial data. We adopt *Manifold Mixup* [25], a state-of-the-art regularizer, to produce the interpolations of these two groups of data, respectively. For each DL system, the *Mixup* result dataset together with the original training dataset are used for retraining. For the second question, we consider the capability of data with different uncertainty patterns. Specifically, we conduct a comparative study on the robustness criteria among data with four uncertainty patterns. According to the preliminary result on CIFAR10, the *NIN* model retrained with data of HL pattern demonstrates the best performance regarding accuracy improvement. The improvement rate is 64% higher than the second best retrained *NIN* with data of LL pattern, while *NIN* models retrained with Deepfool and C&W data achieve negative accuracy improvement. For the robustness against the transfer attacks, the *NIN* model retrained with data of LH pattern is the most effective against the transfer attacks generated on original model, whose average calibration rate on three types of attacks is 92.3%. The evaluation result shows the distinctive capability of data with different uncertainty patterns. More experiments would be performed to achieve a general conclusion.

3 CONCLUSION

In this work, we first perform an empirical study to characterize data inputs of DL systems from the perspective of uncertainty. We then propose a GA-based approach to generate data with more diverse uncertainty patterns. We further investigate the application of the generated data on robustness enhancement. The preliminary results show the effectiveness of the generated data inputs on revealing defects and the usefulness of the diverse data for quality assurance.

Acknowledgement. This work has been supported by the National Natural Science Foundation of China under grant no. 61772038, 61532019, and the Guangdong Science and Technology Department (Grant no. 2018B010107004).

REFERENCES

[1] Nicholas Carlini and David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *IEEE Symposium on Security and Privacy*. 39–57.

[2] Yarin Gal. 2016. *Uncertainty in Deep Learning*. Ph.D. Dissertation. University of Cambridge.

[3] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. 1050–1059.

[4] Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku. 2017. Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960* (2017).

[5] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*. http://arxiv.org/abs/1412.6572

[6] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. 2017. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117* (2017).

[7] Tamir Hazan, George Papandreou, and Daniel Tarlow. 2016. *Perturbations, Optimization, and Statistics*. MIT Press.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[9] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[10] Elmar Haussmann Clement Farabet Kashyap Chitta, Jose M.Alvarez. 2019. Training Data Distribution Search with Ensemble Active Learning. *arXiv preprint arXiv:1905.12737* (2019).

[11] Jinhan Kim, Robert Feldt, and Shin Yoo. 2019. Guiding Deep Learning System Testing Using Surprise Adequacy. In *Proceedings of the 41st International Conference on Software Engineering (ICSE '19)*. 1039–1049.

[12] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. [n.d.]. CIFAR-10 (Canadian Institute for Advanced Research). ([n. d.]). http://www.cs.toronto.edu/~kriz/cifar.html

[13] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial Machine Learning at Scale. *arXiv preprint arXiv:1611.01236* (2016).

[14] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[15] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400* (2013).

[16] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, Jianjun Zhao, and Yadong Wang. 2018. DeepGauge: Multi-granularity Testing Criteria for Deep Learning Systems. In *Proc. of the 33rd ACM/IEEE Intl. Conf. on Automated Software Engineering (ASE 2018)*. 120–131.

[17] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 2574–2582.

[18] Augustus Odena, Catherine Olsson, David Andersen, and Ian Goodfellow. 2019. TensorFuzz: Debugging Neural Networks with Coverage-Guided Fuzzing. In *International Conference on Machine Learning*. 4901–4911.

[19] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. 2016. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *2016 IEEE Symposium on Security and Privacy (SP)*. 582–597.

[20] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. 2017. Deepxplore: Automated whitebox testing of deep learning systems. In *SOSP*. 1–18.

[21] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. 2018. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8571–8580.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[23] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[24] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *ICSE*. ACM, 303–314.

[25] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold Mixup: Better Representations by Interpolating Hidden States. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.), Vol. 97. PMLR, Long Beach, California, USA, 6438–6447.

[26] Jingyi Wang, Guoliang Dong, Jun Sun, Xinyu Wang, and Peixin Zhang. 2019. Adversarial sample detection for deep neural network through model mutation testing. In *Proceedings of the 41st International Conference on Software Engineering*. IEEE Press, 1245–1256.

[27] Xiaofei Xie, Lei Ma, Felix Juefei-Xu, Minhui Xue, Hongxu Chen, Yang Liu, Jianjun Zhao, Bo Li, Jianxiong Yin, and Simon See. 2019. DeepHunter: A Coverage-guided Fuzz Testing Framework for Deep Neural Networks. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2019)*. ACM, New York, NY, USA, 146–157. https://doi.org/10.1145/3293882.3330579

[28] Xiaofei Xie, Lei Ma, Haijun Wang, Yuekang Li, Yang Liu, and Xiaohong Li. 2019. DiffChaser: Detecting Disagreements for Deep Neural Networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 5772–5778. https://doi.org/10.24963/ijcai.2019/800

[29] Weilin Xu, David Evans, and Yanjun Qi. 2017. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155* (2017).

[30] Xiyue Zhang, Xiaofei Xie, Lei Ma, Xiaoning Du, Qiang Hu, Yang Liu, Jianjun Zhao, and Meng Sun. 2020. Towards Characterizing Adversarial Defects of Deep Learning Software from the Lens of Uncertainty. In *ICSE*. ACM.