**Project Proposal**

**BayesFuzz: A Fuzz Testing Framework based on Bayesian Judgment of Neuron Importance for Deep Neural Networks**

COMP7840

Guowei Yang, Bin Duan

Chong Xu 47876312

August 12th, 2024

**Background**

The extensive applications of deep learning network (DNN) grant substantial commercial benefits and significantly improve the quality of human lives. With the widespread deployment of AI technology in safety-critical fields, including autonomous cars [1] and medical treatment [2], the potential risks of undetected errors in DNN systems raise public concerns. A notable example is the fatal crash of Uber’s self-driving car [3], which intensifies the doubts about safety of DNN and increases demand for comprehensive testing for DNN models. The concerns have casted a shadow over its promising potential of DNNs in various fields [4].

There is a surge in research dedicated to the development of DNN testing criteria and frameworks in past seven years [5]. Given the proven effectiveness of existing developed software testing methods, researches like Sun et al. [6] implement the methodologies for DNN testing. However, Pei et al. [7] argue that the properties of DNN system, the internal interaction among neurons, is a significant part of the testing criteria, and develops a new metric, neuron coverage, to enhance the diversity and comprehensiveness of testing cases. The white-box testing framework, DeepXplore [7], is designed to demonstrate the advantages of neuron coverage, followed by Test4Deep [8] and Adapt [9], both extend and refine the algorithm of DeepXplore to improve DNN testing quality. Despite the advancements, the study of DNN testing is still at early stage, and the consensus on testing criteria has not yet been reached. Some researches present counterexamples of neuron coverage, that challenges the reliability of the metric in certain testing contexts [10].

Furthermore, an innovative topic of DNN testing, interpretability[5], raises the interests of experts. The study aims to make the mechanism and decision-making procedures of DNN understandable to humans. The topic provides a novel perspective for DNN testing and contributes to the development of frameworks focusing on the unique properties of DNNs. Detailed discussions of current research in DNN testing will be discussed in the following sections.

**Literature Review**

Comprehensive testing for software requires a well-defined testing metric and diverse testing cases to detect corner cases. This is especially critical to DNN systems, which are deployed in various fields tied to public safety [1] [2]. Even rare bugs happening in extreme situations can lead to catastrophic consequences. Current research about DNN testing focuses on several areas: verification, interpretability, testing criteria, and testing framework. The following sections summarize papers contributing to DNN testing from different prospectives.

**Verification and Interpretability**

The intricate structure and mechanisms of neural network hinders the complete analysis of the decision-making process[11]. Consequently, researchers seek methods to analyze the DNN systems using human-understandable language and traditional software testing techniques. Boolean satisfiability (SAT) and satisfiability modulo theories (SMT) solving techniques are employed to analyze the properties of DNN networks [12]. These approaches explore the non-linear activation functions of the neurons, asserting that the safety of abstract model of DNN can be transformed to its concrete implementation [5]. In addition, Black-box testing method is employed to analyze the properties of DNN systems and evaluate factors that significantly impact the testing results. Liang et al. [13] investigate the influence functions on DNN model’s prediction results, observing the effects on the learning procedure.

The study of theoretical verification and black-box testing employes the traditional software testing techniques to achieve effective verification of robustness and reliability of DNN applications. However, the ignorance of internal process of DNN systems leads to a lack of comprehensiveness in practical context [5].

**Development of White-box Testing**

The significant difference between white-box and black-box testing is the former tests the software by analyzing the internal structure and interactions among parameters during testing. While white-box testing implementation is more challenging for DN, it is a promising field for testers to fully verify and understand the reliability of DNN systems, ensuring the safety of the deployment.

DeepXplore [7] is the first white-box framework that aims to test the models through examining their internal structure. Pai et al [7] proposes a testing criteria, neuron coverage, and integrates it into DeepXplore to improve the diversity of the tests cases. The framework implements a joint optimization function to generate differential behavior between DNN models and maximizes the number of active neurons. One of the noticeable features of the framework is that it conducts an unsupervised learning process by integrating multiple DNNs with similar functions and comparing their results to verify correctness. Test4Deep [8] and Adapt[9] extend the algorithm, and improve the framework with refined test case generation procedure and innovative neuron selection strategies. Test4Deep identifies one of the limitations of DeepXplore is that it requires multiple DNNs for verification. To address the problem, Test4Deep designs a gradient-ascent method to enable single DNN testing [8]. Adapt introduces a transformation method to convert each neuron to a multi-dimension feature, which quantifies the impacts of neurons and provides an innovative perspective on the structure of DNNs.

It is important to note that Sun et al [6] argue that neuron coverage is not a rigorously reliable metric for DNN testing, because it ignores the relations existing in the DNN network. The critique contributes to the development of this project, BayesFuzz, which aims to integrate the knowledge of relations of layers within DNN into the white-box testing procedure.

**Motivation**

Existing studies about white-box DNN testing generally recognize the reliability and effectiveness of neuron coverage as the testing metric, while some experts argue the metric fails to account for the relationships among neurons [6]. Although some revised testing criteria have been proposed based on traditional software MC/DC testing methods [6], there is no state-of-the-art framework that evaluates the significance of single neuron during learning process and leverages the information to improve the efficiency of DNN systems.

This project aims to develop a DNN white-box testing framework, based on the existing theory of Bayesian optimization [14]. The goal is to translate the theoretical principles of Bayesian learning into practical implementation within DNN models. The framework could dynamically identify the influences of each neuron during training phase and adjust the parameters of the activation function of neurons to either amplify or mitigate their impacts on the prediction outcomes. From a broader perspective, this research aims to underline the significance of correlations among parameter groups in DNN testing and contribute to the evolution of DNN white-box testing criteria through empirical validation.

**Project Plan**

This project plans to implement the testing framework with TensorFlow [15] and Keras [16] on a laptop with NVIDIA GeForce RTX 2070 Max-Q GPU and Intel(R) Core (TM) i7-9750H CPU. Two datasets, MNIST [17] and ImageNet [18], and six DNN pretrained models, LeNet-1, LeNet-4, LeNet-5, VGG-16, VGG-19, and ResNet50 are selected for the testing.

The primary objective of this project is to demonstrate the efficiency and effectiveness of BayesFuzz, so it will be compared with the existing frameworks, DeepXplore [7] and DLFuzz [19], to prove the advantages of Bayesian learning in DNN testing. The three frameworks will be tested with the datasets and DNN models, and the performance will be based on neuron coverage, the number of generated test cases, and the minimization of added perturbations to the original inputs. By benchmarking BayesFuzz against other frameworks, this project aims to highlight the advantages of Bayesian learning for DNN testing.

**Bayesian Optimization**

Bayesian optimization lays the core theoretical foundation of this project. It is a global optimization strategy that describes the behavior pattern of test inputs with a probabilistic model, generally Gaussian Process, and further decisions will be made based on the model to improve the input function. In this project, the objective function in terms of the network’s predictive accuracy is given in the following part, where x represents the configuration of neurons and is the test dataset.

In this project, a Gaussian Process will be used to model the neuron’s behavior with the following mean function m(x) and covariance function k (x, x’).

The update functions of the GP posterior are listed below.

At the context of this project, it is expected that above equations can model the behavior patterns of groups of neurons in DNN models by collecting data from testing procedure, and the tester would identify critical neurons that contribute most for the prediction results based on the model. After filtering out the critical neurons, the research team would develop algorithm to adjust the parameters of the activation function [11], like weight or bias, to discriminately modify the influences of neurons during testing.

**Datasets and Models**

MNIST and ImageNet are widely-adopted datasets for DNN testing. These two datasets are specifically designed for image and pattern classification tasks.

MNIST consists of 60,000 examples of training set and 10,000 examples of test set. The data type is handwritten digits with a resolution of 28x28 pixels. The labels of the data are the digits 0-9. The format of the data is compatible with current software. The resource can be accessed from the homepage.

LeNet is one of the earliest convolutional neural networks (CNNs) used for image classification tasks[20]. It was developed by Yann LeCun to work on handwritten digit recognition [20]. This model has various versions with different complexities. LeNet-5 has the most layers and feature maps in its CNN structure, and therefore possess higher complexity and more powerful prediction accuracy than LeNet-1 and LeNet-4 [21]. The three versions of LeNet will be used for this project to make predictions on MNIST dataset.

ImageNet consists of 14,197,122 images covering a diverse range of real-life objects. The dataset has a hierarchical structure of images with full resolution [22]. The diversity and complexity of the contents make ImageNet a valuable test resource for this project. The contents in the MNIST only focus on digit while ImageNet involves more detailed images which are closely related to real-life applications. The images also have labels for deep learning. The resource can be accessed from the homepage, too.

VGG16 and VGG19 are two versions of Visual Geometry Group (VGG) neural networks with 16 layers and 19 layers [23]. The architectures are designed for DNN testing in ImageNet dataset. The authors develop the DNN based on ConvNet architectures, achieving a more efficient prediction tools. Those architectures are built-in functions in Keras package, which grants convenience for the experiment preparation. ResNet-50 [24] is another architecture that will be used for testing on ImageNet dataset. It is a version of Residual Networks (ResNet) family[24] and it has 50-layer DNN structure. The ResNet family uses Residual Blocks to solve the performance degradation problem in DNNs and they exhibit great performance on image classification tasks. The model can be accessed from GitHub page.

**Testing Criteria**

The testing criteria for the experiment consists of neuron coverage [7], test case generation efficiency and quality. Neuron coverage is a popular testing metric developed by Pei et al. [7], and it is widely employed by researchers of DNN white-box testing [5]. The metric measures the portion of active neurons during the test process. A higher neuron coverage represents more neurons involves in the prediction procedure, leading to the increase in diversity of outputs [7]. Although the reliability of the metric is doubted [10], it is still a valuable testing criterion to evaluate the diversity of the test cases. To compensate for the gap, test case generation speed and quality is included in the testing criteria. The frameworks used in the experiment will be tested on the number and quality of generated test cases at a limited period. The quality of generated test cases can be measured by human eyes and the portion of perturbations added to the original input. If the percentage of perturbations is low and the generated images cannot be easily distinguished by human-eyes from the original input, the test cases quality would score a high mark.

**Milestones**

The section outlines the weekly plan and milestones for the project.

Week 5: Read papers about Bayesian optimization and conducting existing DNN testing methods on the laptop.

Implementing the Bayesian learning method on MATLAB is expected for the experiment design. There are various existing DNN testing open-sources codes on GitHub, while many uses outdated packages so it may require more time and effort on completing the experiments.

Week 6: Apply the Bayesian learning algorithm in the convolution neural network in Python and observe the results. Prepare for the seminar.

The actual implementation of Bayesian optimization on the DNNs may take much time on adjusting the parameters and fitting the algorithm with the existing models. It is expected that the optimization method can work on any type of testing framework of DNNs and demonstrate some results. It is also necessary to prepare a presentation, slides and notes for the seminar in week 7.

Week 7: Integrate the Bayesian methods with the existing white-box framework. Participate seminar.

This week involves seminar participation, so there is no rigid requirement of completing tasks. It is expected that this week continues the experiment from last week to enable the implementation of Bayesian optimization. If the experiment proceeds successfully, the team will consider integrating the optimization methods with existing white-box testing frameworks.

Week 8-10: Conduct the experiments and record the results. Get feedback from the supervisor for further modification. Prepare for conference paper, poster and demonstration.

The main experiment procedure is to achieve the Bayesian learning method on existing white-box testing frameworks to demonstrate the advantages of the optimization algorithm. It is expected that the selection of testing metric and parameters would take most of the experiment phase and substantial data collection would also be considered. The team would conduct multiple times of experiment to verify the reliability of the testing results in these weeks. If the experiment proceeds successfully, the team will use some adversarial generator to obtain some extreme inputs for the framework, to test its robustness and accuracy with complex configurations.

Week 11-12: Prepare poster, demonstration and write the thesis and complete the thesis.

These two weeks will be used for the completing of assessment items for the course, debugging the program and preparing for demonstration of the thesis project. It is expected that the poster would include the necessary mathematical equations, the basic mechanism of the framework and the experiment results.

**OHS Risk and Ethic Assessment**

**Risk:**

The project only involves the use of personal laptops with open-source resources. If there are extra requirements for hardware, low-risk laboratory covered by OHS laboratory rules in the University of Queensland will be considered as the testing area.

**Ethic:**

The datasets and packages used in this project are all open-source resources and available for academic usage.

**Generative AI Usage**

ChatGPT is used to revise the grammar and wording of this document [25].

**Reference**

[1] Y. Cui *et al.*, “Deep Learning for Image and Point Cloud Fusion in Autonomous Driving: A Review,” 2020, doi: 10.48550/ARXIV.2004.05224.

[2] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, “Deep learning for healthcare: review, opportunities and challenges,” *Briefings in Bioinformatics*, vol. 19, no. 6, pp. 1236–1246, Nov. 2018, doi: 10.1093/bib/bbx044.

[3] “Uber’s self-driving operator charged over fatal crash,” *BBC*, Sep. 17, 2020. [Online]. Available: https://bbc.com/news/technology-54175359

[4] J. M. Zhang, M. Harman, L. Ma, and Y. Liu, “Machine Learning Testing: Survey, Landscapes and Horizons,” *IEEE Transactions on Software Engineering*, vol. 48, no. 1, pp. 1–36, Jan. 2022, doi: 10.1109/TSE.2019.2962027.

[5] X. Huang *et al.*, “A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability,” *Computer Science Review*, vol. 37, p. 100270, Aug. 2020, doi: 10.1016/j.cosrev.2020.100270.

[6] Y. Sun, X. Huang, D. Kroening, J. Sharp, M. Hill, and R. Ashmore, “Testing Deep Neural Networks,” 2018, *arXiv*. doi: 10.48550/ARXIV.1803.04792.

[7] K. Pei, Y. Cao, J. Yang, and S. Jana, “DeepXplore: automated whitebox testing of deep learning systems,” *Commun. ACM*, vol. 62, no. 11. Association for Computing Machinery, pp. 137–145, 2019.

[8] J. Yu, Y. Fu, Y. Zheng, Z. Wang, and X. Ye, “Test4Deep: an Effective White-Box Testing for Deep Neural Networks,” in *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, Aug. 2019, pp. 16–23. doi: 10.1109/CSE/EUC.2019.00013.

[9] S. Lee, S. Cha, D. Lee, and H. Oh, “Effective white-box testing of deep neural networks with adaptive neuron-selection strategy,” *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*. Association for Computing Machinery, Virtual Event, USA, pp. 165–176, 2020. [Online]. Available: https://doi.org/10.1145/3395363.3397346

[10] L. Ma *et al.*, “DeepGauge: Comprehensive and Multi-Granularity Testing Criteria for Gauging the Robustness of Deep Learning Systems,” Mar. 2018.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: https://books.google.com.au/books?id=Np9SDQAAQBAJ

[12] L. Pulina and A. Tacchella, “An Abstraction-Refinement Approach to Verification of Artificial Neural Networks,” in *Computer Aided Verification*, vol. 6174, T. Touili, B. Cook, and P. Jackson, Eds., in Lecture Notes in Computer Science, vol. 6174. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 243–257. doi: 10.1007/978-3-642-14295-6\_24.

[13] P. W. Koh and P. Liang, “Understanding Black-box Predictions via Influence Functions,” in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., Proceedings of Machine Learning Research: PMLR, 2017, pp. 1885--1894. [Online]. Available: https://proceedings.mlr.press/v70/koh17a.html

[14] T. Chen *et al.*, “An Efficient Sharing Grouped Convolution via Bayesian Learning,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 33, no. 12, pp. 7367–7379, Dec. 2022, doi: 10.1109/TNNLS.2021.3084900.

[15] *TensorFlow*. [Online]. Available: https://tensorflow.google.cn/

[16] *Keras*. [Online]. Available: https://github.com/keras-team/keras

[17] Y. LeCun, C. Cortes, and C. J. C. Burges, “THE MNIST DATABASE.” [Online]. Available: https://yann.lecun.com/exdb/mnist/

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL: IEEE, Jun. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.

[19] J. Guo, Y. Jiang, Y. Zhao, Q. Chen, and J. Sun, “DLFuzz: differential fuzzing testing of deep learning systems,” *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. Association for Computing Machinery, Lake Buena Vista, FL, USA, pp. 739–743, 2018. [Online]. Available: https://doi.org/10.1145/3236024.3264835

[20] Y. L. Cun *et al.*, “Handwritten digit recognition with a back-propagation network,” in *Advances in neural information processing systems 2*, Morgan Kaufmann Publishers Inc., 1990, pp. 396–404.

[21] Y. Lecun *et al.*, *Comparison of learning algorithms for handwritten digit recognition*. 1995.

[22] “ImageNet Homepage.” [Online]. Available: https://image-net.org/index.php

[23] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” 2014, *arXiv*. doi: 10.48550/ARXIV.1409.1556.

[24] P. Potrimba, “What is ResNet-50?” [Online]. Available: https://blog.roboflow.com/what-is-resnet-50/

[25] *ChatGPT*. (2024). OpenAI. [Online]. Available: https://chatgpt.com/