Australia's National
Science Agency

Framework of AI/ML Dependency Assessment and Penetration Testing

Hongsheng Hu, Data61

Jason Xue, WP4 lead, Data61

24 January 2023

Contents

1 Project Abstract

Project title:

Framework of AI/ML Dependency Assessment and Penetration Testing

Project Start Date: 03/11/2022 **Projected Finish Date:** 01/05/2024

Project Abstract:

Model testing and evaluation for deep learning is a critical step in the field of artificial intelligence, which has somehow been overlooked but is fundamental for software and system success such as in cyber-physical systems. As an integral component for deep learning software, the most common research method for practical and meaningful testing includes the training and testing partition for model evaluations. While a general goal of such testing scheme is for the robustness of deep learning software, this project proposes a system for evaluating various types of deep learning models based on multiple testing indicators and visualization techniques, which means that the system will not only quantitatively evaluate the model against specific indicators on the input data set, but will also evaluate the model in greater depth from the perspectives of robustness, generalization, practical utility, and model interpretability. Also, it will provide a test bed for the model's viability in an industrial setting. It should serve two primary purposes. The first is that the model can be tested and evaluated using multiple evaluation indicators and adversarial attacks. The second advantage is that these evaluation indicators can be dynamically visualised with the designed goal.

Project Objectives:

Development of a deep learning model assessment system based on adversarial attack, with multiple evaluation metrics. It can perform model evaluation and robustness testing on different types of data, such as images and text, and different trained deep learning models, such as RNNs and CNNs, thus assisting researchers with a more efficient deep learning model testing and evaluation.

2 Project Description & Scope

Project Title:

Framework of AI/ML Dependency Assessment and Penetration Testing

Product Characteristics and Requirements:

1. It needs to have a user-friendly interface.
2. It needs to support multiple types of data for testing.
3. It needs to support evaluation testing of multiple models (model robustness and parameter coverage).
4. It needs to be modified for the data parameters.
5. It requires visualization of the corresponding parameters.
6. It requires visualization of the structure of the model.
7. It requires visualization of the model testing and evaluation results.

Summary of Project Deliverables:

Project management-related deliverables:

1. Create relative deliverables: allocate the roles of team members; determine project goals (evaluating and testing AI/deep learning models); determine the scope of the project; determine the project technology (React, python, Grad-CAM, attention).
2. Cohort of supervisors and team leader: The project must be supervised by a group of members who are familiar with the entire system and capable of fully considering where future expansions and upgrades may be required to the final system.
3. Define the project and process: formulate a detailed development plan (front-end, back-end, algorithm, tester) and project milestones.
4. Achieve the expected result: Determine whether the project meets internal or external expectations.

Product-related deliverables:

1. Start-up phase:

   Extensive team discussions to determine system requirements and identify core and additional requirements. and identify project technologies.

2. Design Phase:

   Need to design the prototype and user operation UI

   Final output: product prototype file, UI interface

3. Development phase:

   The team leader needs to manage the entire project, schedule people, and ensure that tasks are completed on time and on schedule. Developers and testers are required to carry out development tasks according to the plan.

   Outputs: test versions and release candidates.

4. Evaluation and feedback phase:

   Candidate versions are provided to experts for evaluation and feedback, and rapid iterative changes are made in response to the feedback.

   Final output: full version and documentation.

Project Scope:

This project aims to develop a deep learning model testing and evaluation system to assist stakeholders and researchers in evaluating their trained models more intuitively and conveniently.

- To develop a user-friendly front-end user page, it needs to support image and text type data upload.
- Develop a visual display page for model evaluation indicators.

- Parameters can be selected when testing the model. For diagrams, it can be to add noise and flip operation; Synonym substitutions can be made for text.
- The accuracy rate, recall rate, F1 score, confusion matrix, ROC curve, and AUC will be used to evaluate the classification task related models.
- For the semantic segmentation model, pixel accuracy, average pixel accuracy, average intersection ratio, frequency weight intersection ratio, and Dice coefficient will be used to evaluate species evaluation indexes.
- The target detection model will be evaluated using IoU, accuracy, recall rate, average accuracy, and average class accuracy.
- Six indexes of accuracy rate, accuracy rate, recall rate, F1 score, Exact Match, and MRR will be provided for the text classification model.
- For the Text2Text model, five indexes including BLEU, ROUGE, NIST, METEOR, and TER will be provided for evaluation.
- Test and evaluate the parameter coverage of the model.
- By means of adversarial attack, FGSM, BIM, DIM, TIM, SINI, and other methods were used to test the robustness of the model, and advGAN will be used to test multiple data.
- For the image related models, SM, BIG and Grad-CAM will be used for visualization.
- Visualize text type tasks using attention.

Process Requirements:

- User-friendly and well-design interface
- Completion of the project within the specified time frame
- Complete all core functions
- Provides two robustness test metrics
- Provides at least one model test based on adversarial attack

Environmental Constraints:

- Python
- Vue

Project License:

- BSD

Key Schedule Milestone:

1. Project planning stage: November 03, 2022
2. AI/ML detection: Feb 01, 2023
3. AI/ML dependency assessment: May 01, 2023
4. AI/ML penetration testing: August 01, 2022

5. Enhanced AI/ML penetration testing: November 01, 2023
6. Risk assessment: Feb 01, 2024
7. Enhanced risk assessment: May 01, 2024

3 Literature Review

Model testing and evaluation are critical components of machine learning software, especially for deep learning systems. But the practice of such testing and evaluation of deep learning models have been frequently overlooked by researchers and practitioners. There has been a series of mature software system testing and evaluation processes in the traditional software development process. However, in the field of machine learning, most of the time there will be only test set data being available for the evaluation of the trained model, which is done by calculating the accuracy of the model and other indicators. This evaluation method does not adequately assess the model's stability and generalization. At the moment, some researchers have recognized this issue and are conducting research. To assess the model's robustness and parameter coverage, methods such as adversarial attacks are proposed.

Pei et al. (2017) point out that in the traditional software development process, there is code coverage to evaluate software tests, which is similar to the coverage of neural networks in the neural network. By assigning a value to each neuron, if it is lower than the set threshold, it indicates that the neuron is not activated. Furthermore, the authors wonder if the model is stable across different data sets. To assess the model's stability, the author proposes training the neural network model several times and then comparing the similarity of the model results. If the similarity is high, it means that the neural network model is stable. Finally, by combining the two concepts mentioned above, the author defines two corresponding loss functions that can evaluate the model's stability and parameter coverage of the model respectively.

Wang et al. (2022) illustrate a method to optimize model data input, which aims to solve the problem of model forgetting. When training the NLP task, the author proposed that the input from each task be combined with the sample. First, the input task is defined, and it returns Yes if it meets the specified conditions and return No otherwise. An explanation is given for each return, telling the model why it is correct or incorrect, which is similar to how humans learn. In a nutshell, the goal of this paper is to train similar tasks together. In this paper's experiment, the method of training similar tasks together produced better results than training alone.

Wang et al. (2021) point out some concepts of the robustness of evaluation test models that may be applicable to a wide range of machine learning research fields and use these concepts to select data sets more suitable for countering aggression. As an example, consider DL Robustness. The definition is that if two input samples are relatively similar, then the output is also relatively similar after passing through the model, indicating that the model is relatively robust. There is also Empirical Robustness, which is defined as an attack on a model, and if the model's output remains stable under attack, the model is said to be robust. This paper proposes Zero-Order Loss (ZOL) for Loss, which refers to the similarity of a sample in the test sample to the true value. Its defect is that evaluating the entire model with a single test sample is difficult. As a result, the author suggested First-Order Loss (FOL). The most significant difference between FOL and ZOL is that test cases are compared in a relatively uniform manner by optimizing seed in the gradient direction, which can measure the model's maximum loss and help researchers choose data sets more suitable for fighting attacks.

Wang et al. (2021) point out that adversarially-robust deep networks used for image classification are more decipherable because their feature identifications tend to be sharper and are more focused on the objects related to the image's ground-truth class. The model's input gradients around data points will more closely align with the decision boundaries' normal vectors when they are smooth, which is how we demonstrate that smooth decision boundaries play a significant role in this improved interpretability. The research shows that determining the appropriate decision boundary improves the model's interpretability. The normal vector in linear models is perpendicular to the decision boundary, whereas the normal vector obtained by finding the gradient in ReLU models may be a fuzzy result, such as an extension of the decision boundary, which may cause the result to fail. The degree of response to each factor is represented by attributes. To simulate the absence of causes, deep network attributions require a baseline. We must select the most interpretable factor from among the many inputs. It is possible to choose a baseline with asymptotic 0 prediction results for many deep networks. When interpreting the attributive results, the integral gradient method can ignore the baseline while assigning attributive results based solely on the input.

4 System Design

The primary goal of the first version of this system is to validate the implementation effect of core functions, so only the core functions in the system that need to validate the theory's feasibility are planned to be developed, which are data uploading, model uploading, parameter customization, model robustness testing, parameter coverage testing, model visualization, and generating evaluation reports.

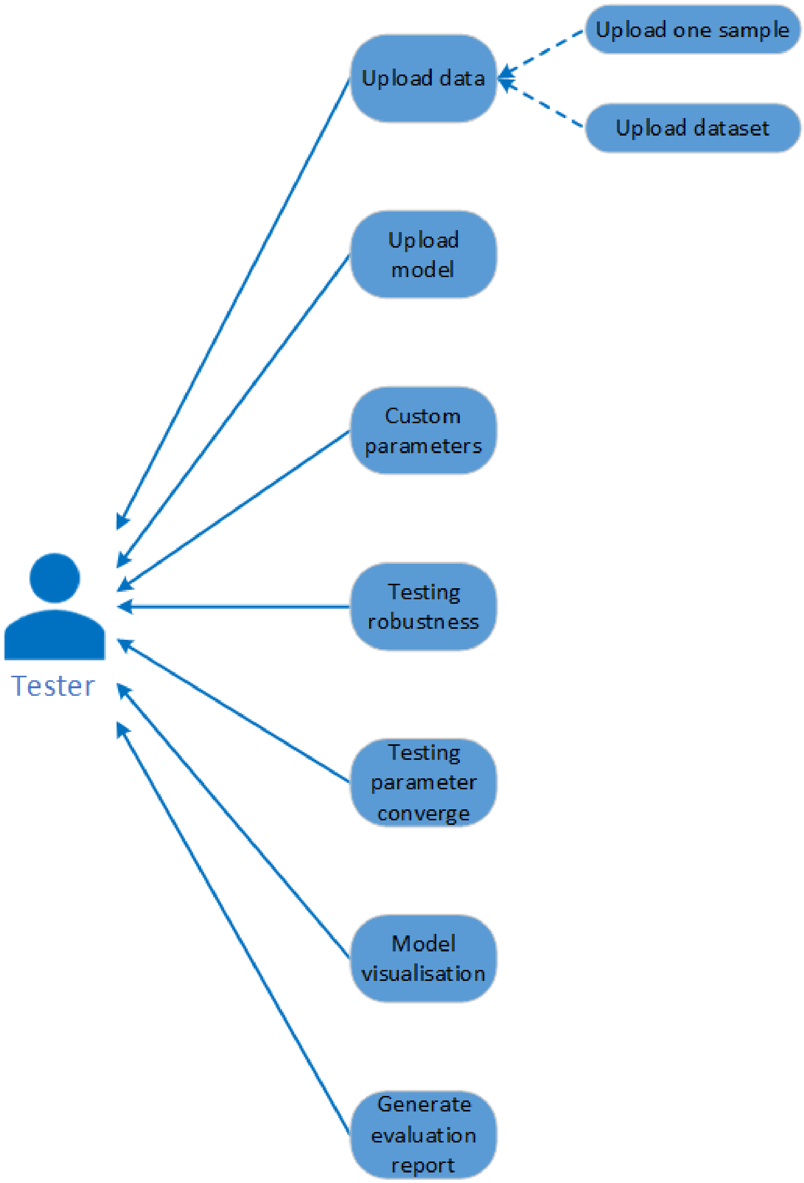

Figure 1: Use case diagram of the system.

5 Quality management

5.1 Define quality standards

1. User interface should be simple and user-friendly
2. User interface supports multiple data types for upload
3. The system can be data enhanced for many types of data
4. The system supports multiple types of model tests to assess its robustness
5. The system supports multiple types of model tests to evaluate their parameter coverage
6. The system supports the visual presentation of multiple models
7. The system can generate evaluation reports for the model

5.2 Measure Project quality

1. Clear, unambiguous and meaningful text prompts on the user interface
2. A sense of feedback in the operation of the user interface
3. Adequate error indication on the user interface
4. Simple and clear user interface UI design
5. User interface to upload data with drag and drop support
6. The system can enhance image data by cropping, rotating, masking, modifying contrast, etc.
7. The system can enhance text data by replacing close synonyms
8. The system supports the adoption of multiple evaluation indicators
9. The system allows visualisation of the model
10. The system can provide assessment reports via PDF files and Boards etc.

5.3 Quality Assurance

5.3.1 Analyse project quality

Function test

- Black box testing to verify the operation of the system for vulnerabilities or errors.
- Evaluate the effectiveness of the test model with a known dataset and model validation.

Unit test

- The code coverage of unit tests needs to be at least 80%.

5.3.2 Improve project quality

Firstly, through the testing and evaluation criteria described above, the development of this project will be carried out in accordance with strict quality management standards, thus ensuring the overall quality of the project. Secondly, as the theoretical technologies used in this system are cutting edge, we will continue to track these technologies and integrate them during the development process. We will also continue to consult with relevant academic and industrial experts and use this feedback to improve the quality of the project if further useful feedback is received.

Reference

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248–255).

Pei, K., Cao, Y., Yang, J., & Jana, S. (2017). DeepXplore: Automated Whitebox Testing of Deep Learning Systems. Proceedings of the 26th Symposium on Operating Systems Principles, 1–18. https://doi.org/10.1145/3132747.3132785

Wang, Y., Mishra, S., Alipoormolabashi, P., Yeganeh Kordi, Mirzaei, A., Arunkumar, A., Ashok, A., Arut Selvan Dhanasekaran, Naik, A., Stap, D., Pathak, E., Karamanolakis, G., Haizhi Gary Lai, Purohit, I., Mondal, I., Anderson, J., Kirby Kuznia, Doshi, K., Patel, M., … Khashabi, D. (2022). Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. arXiv.org.

Wang, J., Chen, J., Sun, Y., Ma, X., Wang, D., Sun, J., & Cheng, P. (2021). RobOT: Robustness-Oriented Testing for Deep Learning Systems. 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 300–311. https://doi.org/10.1109/ICSE43902.2021.00038

Wang, Z., Fredrikson, M., & Datta, A. (2021). Robust Models Are More Interpretable Because Attributions Look Normal. arXiv.org.