Robustness Testing Framework For Neural Network Image Classifier

Duo Li*, Chaoqun Dong and Qianchao Liu
Jiangnan Institute of Computing Technology, Wuxi, Jiangsu, 214083, China
*Corresponding author’s e-mail: lidxz9585@163.com

Abstract. Neural network has made remarkable achievements in the field of image classification, but they are threatened by adversarial examples in the process of application, making the robustness of neural network classifiers face danger. Programs or software based on neural network image classifiers need to undergo rigorous robustness testing before use and promotion, in order to effectively reduce losses and security risks. To comprehensively test the robustness of neural network image classifiers and standardize the test process, starting from the two aspects of generated content and interference intensity, a variety of robustness test sets are constructed, and a robustness testing framework suitable for neural network classifiers is proposed. And the feasibility and effectiveness of the test framework and method are verified by testing LENET-5 and the model reinforced by the adversarial training.

1. Introduction
The robustness of neural network model refers to the stability of the model under various normal and abnormal inputs[1]. At present, the traditional method of robustness testing for system or software is to construct various abnormal inputs and observe their ability to deal with these abnormal inputs. The training data set and test data set of neural network image classifier are only a small part of the sample space, and the selected images are generally characterized by bright features and high image quality, which are difficult to realize in practical application. So the accuracy of the model in the test set is difficult to reflect its robustness. In order to get a more comprehensive test set, it is unnecessary to collect, organize and mark additional test sets, which will greatly increase the test cost. At the same time, the existence of adversarial samples also makes the robustness of neural network image classifier face challenges. Adversarial samples refer to adding tiny and imperceptible disturbances to the image to make the classification error of image classifier. Goodfellow[2] attributed the existence of adversarial samples to the linear characteristics of deep neural network in high-dimensional space. Adversarial samples exist widely in neural networks, and there is no effective way to solve the problem.

There is no standard process and fixed method for that how to test the robustness of neural network image classifier at present. Based on the existing research theories and methods, this paper expands a variety of robustness test sets based on the original test data set, and focusing on the two aspects of constructing robustness test data set and standardizing test process, we propose a robustness testing framework for neural network image classifiers. Experiments on MNIST data set using lenet-5 image classifier model verify the feasibility and effectiveness of the framework and method.
2. Overview
In this section, we mainly introduce the robustness testing framework of neural network image classifier, and explain the main parts of the framework.

2.1. Image geometry operation
The robustness testing framework of neural network image classifier is shown in Figure 4. In this framework, the robustness of the model is tested by expanding the test set. The framework is mainly divided into two modules: robustness test based on multi-directional generated content and robustness test based on single direction interference intensity. Firstly, the robust scene requirement of image classifier is analyzed to determine the test content. The robustness test based on multi-directional content mainly considers a variety of conditions that may affect the robustness of image classifier, and tests the classifier by building a test set group $T'$ that simulates these conditions. The robustness test set is built in three ways: Based on the basic test set $T$, the test set $T1'$ is built by geometric operations, the test set $T2'$ is built by changing the image quality, the test set $T3'$ is built by generating adversarial sample algorithms. The robustness test based on single direction interference intensity mainly considers that in some special scenarios, it may be necessary to know more about the robustness of the classifier in a specific environment or interference. Firstly, the test items and interference intensity range are determined, and then the corresponding test set is generated through a variety of interference intensity settings to build the test set group $T''$. Finally, the test set generated in the above way is given for the image classifier to recognize, and the accuracy and other information are obtained. According to the list of robust ability rank, the robust ability of the image classifier is ranked, which provides a reference for the analysis and evaluation of the robust ability of the image classifier.

2.2. Robust scenario requirement analysis
Robust scenario requirement analysis is to analyze the application scenarios of neural network image classifier, determine the test requirements, and then determine the content to be tested. Three application scenarios are mainly considered: the simple application scenario, general application scenario and security application scenario. In simple application scenarios, considering the image shooting angle, distance and other factors, the robustness of the neural network image classifier for some image geometric operations needs to be tested; In general application scenarios, random noise will be added in the process of image transmission, storage and conversion, so it is necessary to detect the robustness of neural network classifier under changing image quality; In the field of security applications, classifiers may be threatened by adversarial samples built by malicious attackers. It is necessary to detect the robustness of neural network image classifiers under adversarial sample attacks. In addition, it may be necessary to understand the robustness of image classifier under some operation, such as some adversarial sample attack, then it is necessary to set different interference intensity adversarial samples to test the robustness of image classifier under some adversarial sample algorithm attack.

2.3. Building test data sets
The core of neural network image classifier robustness testing framework is to build a robustness test set. By building a variety of test sets, the robustness of image classifier is tested. In the robustness test based on multi-directional generated content, referring to the three application scenarios described above, the test set is generated in three ways:

1) **Geometric operation of image:** geometric operation of image to generate test set, which mainly includes translation, rotation, enlargement or shrinking, affine transformation, flipping, etc.

2) **Change the image quality:** change the image quality to generate test set, mainly including adding gaussian noise, adding salt and pepper noise, blur processing, Converting color image to gray image, sharpening processing, changing brightness, changing contrast, etc.
3) **Generate adversarial samples:** use adversarial sample algorithm to generate test set, mainly including black box algorithm and white box algorithm.

Table 1, 2, 3 show some image geometric operations, image quality changing operations and adversarial sample generating operations selected and defined in this paper.

| Operation name | Specific operation                                    |
|----------------|-------------------------------------------------------|
| Geom1          | Enlarge (or shrink) the objects in the image appropriately |
| Geom2          | Shift the image by a certain distance                 |
| Geom3          | Rotate the image clockwise (or counterclockwise) by a certain angle |
| Geom4          | Appropriate affine transformation of the image         |
| Geom5          | Flip the image                                        |

| Operation name | Specific operation                  |
|----------------|-------------------------------------|
| Qua1           | Add Gaussian noise to the image     |
| Qua2           | Add salt and pepper noise to the image |
| Qua3           | Blur the image (Gaussian Blur)      |
| Qua4           | Change image brightness             |
| Qua5           | Convert color image to gray image   |
| Qua6           | Change image contrast               |
| Qua7           | Sharpen the image                    |
Table 3. 8 algorithms to generate adversarial samples

| Operation name | Specific operation |
|----------------|--------------------|
| Adv1           | Use FGSM [2] to generate test set |
| Adv2           | Use JSMA [3] to generate test set |
| Adv3           | Use DEEPPOOL [4] to generate test set |
| Adv4           | Use I-FGSM [5] to generate test set |
| Adv5           | Use MI-FGSM [6] to generate test set |
| Adv6           | Use ONE-PIXEL [7] to generate test set |
| Adv7           | Use C&W [8] to generate test set |
| Adv8           | Use L-BFGS [9] to generate test set |

2.4. Robust capability classification

By building test sets, the accuracy of neural network image classifier on different test sets can be obtained, then we can measure the robustness of the model. Considering that the robustness of the model should be based on its performance on benchmark set, the relative error rate (Rer) is defined:

\[
Rer = T_{\text{acc}} - T'_{\text{acc}}
\]

As an index to describe the robustness of classifier, \( T_{\text{acc}} \) is the accuracy of the classifier on the benchmark set, \( T'_{\text{acc}} \) is the accuracy of the classifier on the robustness test set. At the same time, in order to intuitively reflect the robust ability of the classifier, the robust ability of the classifier is ranked according to the relative error rate, as shown in Table 4.

Table 4. Robust ability rank

| Robustness level | Very good | Good | Common | Poor | Very poor |
|------------------|-----------|------|--------|------|-----------|
| Symbol           | *****     | **** | **     | *    |           |
| Range            | Rer≤0.03  | 0.03<Rer≤0.1 | 0.1<Rer≤0.2 | 0.2<Rer≤0.5 | Rer>0.5 |

3. Experiments

3.1. Experiments setup

In this section, we will demonstrate the operation process of the framework through specific experiments, and verify the feasibility and effectiveness of our proposed method.

This paper selects the classic neural network model LENET-5 as the experimental object, and uses the handwritten digits data set MNIST as the experimental data set. LENET-5 is a model with a 7-layer network structure, including a convolutional layer, a pooling layer, and a fully connected layer. Although the network structure is simple, it can achieve very high accuracy on the handwritten MNIST data set. The MNIST data set is composed of 250 handwritten digits from different people. Each sample is a 28*28 pixel grayscale picture, of which 60,000 are used as the training set and 10,000 are used as the test set.

Zheng et al. [11] proved that by adversarial training, the ability of the model to resist attacks from adversarial samples can be improved, So as a comparison, this article prepared two models. Model M1 and Model M2 are both image classifier models built based on LENET-5. The M1 and M2 are both trained on the MNIST dataset, and M2 is reinforced by adversarial training. The accuracy of M1 on the test set is 0.987, and the accuracy of M2 on the test set is 0.971.

Experimental environment: cpu: i7-9750H, memory: 16g, gpu: RTX2060, operating system: ubuntu18.04, learning framework: TensorFlow 2.3.0.

3.2. Robustness test experiment based on multi-direction generated content

This experiment mainly tests whether the method proposed in the framework can effectively find the defects of the robustness of the image classifier, and then evaluate the comprehensive performance of the robustness of the image classifier.
In the last section, three methods for building robustness test data sets are proposed. There are 20 methods in total, which can be used as a reference for generating robustness test sets. We select representative methods in each construction method to generate the test set required in the experiment. The specific selection method is shown in Table 5. Considering that some adversarial algorithms need multiple iterations to generate test data, which consumes a long time. In order to take into account the time and test set size, this paper decides to use the above methods to generate 1000 test data sets each, in which the original image is 10000 from MNIST Selection of test data sets.

Use the basic test set to generate the data set of the selected method in Table 5. Considering that some adversarial algorithms need multiple iterations to generate test data, which consumes a long time. In order to take into account the time and test set size, this paper decides to use the above methods to generate 1000 test data sets each, in which the original image is 10000 from MNIST Selection of test data sets.

Table 5 Operations selected to generate test case

| Generation method | Test set name | Operation name | Specific operation |
|-------------------|---------------|----------------|-------------------|
| Geometric operations | T11 | Geom1 | Shrink the size of the image to 0.8 times the original image |
| | T12 | Geom2 | Shift the image 5 pixels to the right |
| | T13 | Geom3 | Rotate the image clockwise by 15° |
| Change image quality | T21 | Chge1 | Add Gaussian noise to the image |
| | T22 | Chge2 | Add salt and pepper noise to the image |
| | T23 | Chge3 | Blur the image |
| Adversarial sample | T31 | Adv1 | Use FGSM algorithm to generate adversarial test samples |
| | T32 | Adv2 | Use DEEPFOOL algorithm to generate adversarial test samples |
| | T33 | Adv6 | Use ONE-PIXEL algorithm to generate adversarial test samples |

Table 6. Interference intensity settings

| Test set name | Disturbance size(ε) | T1'' | T1'' | T2'' | T2'' | T3'' | T3'' | T4'' | T4'' | T5'' | T5'' | T6'' | T6'' |
|---------------|---------------------|------|------|------|------|------|------|------|------|------|------|------|------|
|               |                     | 0.05 | 0.10 | 0.15 | 0.20 | 0.25 | 0.30 | 0.35 | 0.40 | 0.45 | 0.50 |

3.3. Robustness test experiment based on one direction interference intensity

The previous part of the experiment is aimed to test and analyze the comprehensive performance of the robustness of the image classifier model. The main purpose of this part of the experiment is to in-depth study the robustness of the image classifier under the same method and different interference intensity settings according to the method proposed in the framework.

The adversarial sample algorithm sets different intensities and number of iterations, and the resulting attack effects are also different. This paper selects the FGSM algorithm in the generation of adversarial samples as the research content, and observes the robustness of the image classifier under different threshold settings. FGSM is an algorithm based on gradient attack. The algorithm can obtain adversarial samples in one step without iteration. The attack expression is:

$$\text{adv}_x = x + \varepsilon \cdot \text{sign} \nabla f(x)$$

Among them, x represents the original image, \( \varepsilon \) is the adjustment coefficient, and the following part represents the gradient. The algorithm mainly controls the disturbance size through the adjustment coefficient, thereby controlling the attack intensity of the algorithm. By setting different thresholds for \( \varepsilon \), this paper generates test sets with different interference intensity, as shown in Table 6. Figure 5 (b) shows some images of the test set generated under different interference settings.

3.4. Results

In the robustness test experiment based on multi-direction generated content, the test sets generated by the above 9 methods are given to for classifier to test. Table 7. shows the accuracy of the classifier on different test sets. Figure 6 shows the relative errors of the models M1 and M2 on each robustness test
Figure 5. Some images of the test case

Table 7. Test results

| Generation method       | Test set name | M1 accuracy | M2 accuracy |
|-------------------------|---------------|-------------|-------------|
| None                    | T             | 0.987       | 0.971       |
| Image geometry perations| T₀₁             | 0.973       | 0.926       |
|                        | T₀₂             | 0.455       | 0.386       |
|                        | T₀₃             | 0.852       | 0.781       |
|                        | T₀₄             | 0.959       | 0.922       |
| Change image quality    | T₁₁             | 0.966       | 0.918       |
|                        | T₁₂             | 0.979       | 0.968       |
|                        | T₁₃             | 0.339       | 0.871       |
| Adversarial examples    | T₂₁             | 0.055       | 0.521       |
|                        | T₂₂             | 0.950       | 0.965       |

Table 8. Robust ability rank

| Operation                  | M1 robustness rank | M2 robustness rank |
|----------------------------|--------------------|--------------------|
| Shrink to 0.8 times        | *****              | ****               |
| Shift 5 pixels to the right| *                  | *                  |
| Rotate clockwise 15°       | ***                | ***                |
| Add Gaussian noise         | *****              | *****              |
| Add salt and pepper noise  | *****              | *****              |
| Blur processing            | ***                | ***                |
| FGSM algorithm attack      | *                  | ****               |
| DEEPFOOL algorithm attack  | *                  | **                 |
| ONE-PIXEL algorithm attack | *****              | *****              |

Figure 6. Relative error rate of different test sets

Figure 7. The accuracy of different interference intensity

Based on the test set generated by the image operation, the models M1 and M2 show good robustness on rotating and shrinking, and poor performance on translation. The accuracy of M1 is 0.455, and the accuracy of M2 is 0.386, both are below 50%. In general, M1’s robustness performance in the these test is better than M1. Based on the test set generated by changing the image quality, the models M1 and M2 both show good robustness. The accuracy of M1 is higher than 0.95, and the
accuracy difference with the basic test set is less than 0.03, and the accuracy of M2 is both above 0.91, the accuracy difference with the basic test set is within 0.05, and the robustness of M1 is still slightly better than that of M2. On the test set based on the generation of adversarial samples, the accuracy of M2 on the FGSM algorithm is 0.871, although compared to the basic test set, there is a certain decrease, to be significantly better than 0.339 of M1; both M1 and M2 perform poorly on the DEEPFOOL algorithm, The accuracy rate of M2 is 0.470, a decrease of 0.5, while the accuracy rate of M1 is only 0.055. It can be seen that M1 has completely lost the ability to work under the attack of DEEPFOOL. In addition, the accuracy of the two on the ONE-PIXEL test set has decreased slightly.

In general, the robustness of M2 under the adversarial sample attack is significantly better than that of M1, which also confirms that adversarial training can indeed improve the model. Robustness under adversarial sample attacks.

In the robustness test experiment based on one direction interference intensity, the experimental results are shown in Figure 7. It can be seen that Model 1 is less robust under the FGSM algorithm attack. When $0<\epsilon<0.25$, the accuracy drops sharply, and when $\epsilon=0.15$, the accuracy has already dropped below. When $\epsilon=0.5$, the accuracy rate drops below 0.1, and the judgment made is meaningless. M2 has strong robustness against FGSM algorithm attack. When $0<\epsilon<0.25$, the accuracy rate declines slowly, the accuracy rate remains above 0.8, and $\epsilon>0.25$, the accuracy rate curve becomes steeper, and the decline speed becomes faster than before, $\epsilon=0.5$, the accuracy rate is 0.352. In short, from the accuracy change curve, under the FGSM algorithm attack, the robustness of the model M2 is significantly stronger than that of the M1.

Through the above experiments, we can draw the following conclusions:

1) Our proposed framework can test the robustness of neural network image classifiers, which includes discovering weaknesses in robustness and characterizing the robustness of a certain aspect.

2) The framework has starability and flexibility, and the test content can be customized according to needs, which can provide a reference for such tests.

4. Related work

Nowadays, with the development and application of neural networks, more and more work has begun to focus on the evaluation and testing of neural network models, and exciting progress has been made. Pei et al.[11] proposed DeepXplore, a white-box testing framework for neural network model systems. For the first time, neuron coverage was used as a metric to study the relationship between neuron coverage and accuracy. Inspired by the idea of MC/DC and combined testing, Sekhon et al.[12], in view of the lack of a clear control flow structure of deep neural networks, making it impossible to apply traditional software testing metrics such as code coverage, they proposed a neural network model. The 2-way coverage criterion and the test input generation method based on the coverage criterion. Guo et al.[13] applied fuzzing testing to neural network testing, and proposed a differential fuzzing testing framework DLFuzz, which is used to guide the neural network system to expose abnormal behaviors. Dwarakanath et al.[14] proposed a metamorphic testing test method for the neural network system for image classification, by designing multiple metamorphic relationships on image data, and then generating a test set to the image classifier to identify, to detect the existence of the neural network system Code defects.

5. Conclusion

This paper starts from the comprehensive detection of the robustness of the neural network image classifier, and aims to effectively measure the robustness of the neural network image classifier and standardize the test process. By generating robustness test sets in multiple methods, the robustness of the neural network image classifier is tested. We propose a test framework suitable for neural network image classifiers, the robustness test based on multi-direction generated content mainly checks the robustness of the model in multiple scenarios, and the robustness test based on single direction interference intensity mainly checks the robustness of the model under different interference
intensities in specific scenarios. The effectiveness and feasibility of this method is verified by testing LENET-5 and its model reinforced by adversarial training. The experimental results show that the framework can effectively find the defects of the robustness of the neural network image classifier, and evaluate the robustness of the classifier model, which can provide a reference for the robustness test of the neural network image classifier.

The robustness problem not only exists in images, but also widely exists in text and speech. In the next step, we will further expand our work to the field of natural language processing and study the robustness problems in the field of natural language processing.

References
[1] Wang, Z., Yan, M., Liu, S., Chen, J. J., Zhang, D. D., Wu, Z., Chen, X. (2020) Survey on testing of deep neural networks. Journal of Software, 31, pp. 1255–1275.
[2] Goodfellow, I. J, Shlens, J., Szegedy, C. (2015) Explaining and harnessing adversarial examples. In: 3rd International Conference on Learning Representations (ICLR).
[3] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., Swami, A. (2016) The Limitations of Deep Learning in Adversarial Settings. In 2016 IEEE European Symposium on Security and Privacy (EuroS&SP), pp. 372-387.
[4] Moosavi-Dezfooli, S., Fawzi, A., Frossard, P. (2016) DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2574-2582.
[5] Huang, X., Kwiatkowska, M., Wang, S., Wu, M. (2017) Safety Verification of Deep Neural Networks," In International Conference on Computer Aided Verification(CAV), pp. 3-29.
[6] Kurakin, A., Goodfellow, I. J., Bengio, S. (2017) Adversarial examples in the physical world," In 5th International Conference on Learning Representations(ICLR).
[7] Carlini, N., Wagner, D. (2017) Towards Evaluating the Robustness of Neural Networks," in 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 39-57.
[8] Su, J., Vargas, D. V., Sakurai, K. (2019) One Pixel Attack for Fooling Deep Neural Networks. In: IEEE Transactions on Evolutionary Computation, 23, pp. 828-841.
[9] Dong, Y. P., Liao, F. Z., Pang, T. Y., Su, H., Zhu, J., Hu, X. L. (2018) Boosting Adversarial Attacks With Momentum" In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9185-9193.
[10] Zheng, P., Song, Y., T. Leung, Goodfellow,I. (2016). Improving the Robustness of Deep Neural Networks via Stability Training. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4480-4488.
[11] Pei, K. X., Cao, Y. Z., Yang, J. F., Suman, J. (2017) Deepxplore: Automated whitebox testing of deep learning systems. In Proceedings of the 26th Symposium on Operating Systems Principles(SOSP), pp. 1-18.
[12] Sekhon, J., Fleming, C. (2019) Towards Improved Testing For Deep Learning. In 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER), pp. 85-88.
[13] Guo, J. M., Jiang, Y., Zhao, Y., Chen, Q., Sun, j. G. (2018) DLFuzz: differential fuzzing testing of deep learning systems. In: 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 739–743.
[14] Dwarakanath, A., Ahuja, M., Silkand, S., Rao, R. M., Bose R. P., Dubash, N., Podder, S. (2018) Identifying implementation bugs in machine learning based image classifiers using metamorphic testing. In: 27th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp. 118-128.