2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)

A Test Cases Generation Technique Based on an Adversarial Samples Generation Algorithm for Image Classification Deep Neural Networks

Song Huang, Qiang Chen*, Zhanwei Hui, Lele Chen, Jialuo Liu, Sen Yang
Command and Control Engineering College
Army Engineering University of PLA
Nanjing, China
e-mail: 359435482@qq.com

***Abstract*—With widely applied in various fields, deep learning (DL) is becoming the key driving force in industry. Although it has achieved great success in artificial intelligence tasks, similar to traditional software, it has defects that, once it failed, unpredictable accidents and losses would be caused. In this paper, we propose a test cases generation technique based on an adversarial samples generation algorithm for image classification deep neural networks (DNNs), which can generate a large number of good test cases for the testing of DNNs, especially in case that test cases are insufficient. We briefly introduce our method, and implement the framework. We conduct experiments on some classic DNN models and datasets. We further evaluate the test set by using a coverage metric based on states of the DNN.**

Keywords-DNN; adversarial samples; coverage metric; test cases generation

I. Introduction

Deep neural networks have been gradually applied in various fields, including speech recognition, image classification, text processing. Especially in the field of image classification, DNNs show great performance advantages, and even surpass the recognition accuracy of the human eyes, to some extent. However, some researches have proved that the deep learning (DL) systems are not as reliable as people think. The image classifiers based on DL can be easily fooled and produce confusing classification results by adding perturbations imperceptible to human eyes to the natural images which are originally classified correctly. These images crafted to cause DL algorithms to misclassify are called adversarial samples. In 2014, Christian Szegedy et al. published a paper in ICLR2014, which officially proposed the concept of adversarial samples for the first time, and gave a specific adversarial samples generation algorithm, L-BFGS; In the subsequent several years, various adverisal sample generation methods have been proposed, for example, FGSM, JSMA, CW and etc.

With the introduction of adversarial samples and the exposure of DNNs' defects, how to test DL software and ensure the quality of DL software has become the focus. Unlike traditional software, programmed with deterministic algorithms by developers, DNNs are programmed by the training data, selected features, and network structures (e.g., number of layers). The internal logic could not be explained, which causes the traditional software testing methods and techniques to be completely ineffective. On the other hand, we need plenty of test cases to accomplish the testing of DL software. So, test cases generation methods are urgently needed especially when the test cases are insufficient. This paper proposes a test cases generation technique based on an adversarial samples generation algorithm for image classification DNNs, which can effectively augment the test corpus.

The main contributions of this paper are as follows: this paper proposes a test cases generation method, which provides a new idea for research of DL software's test cases generation; we then apply a test coverage metric[1] based on the "states" of neural network to evaluate the generated test set; the proposed method is applied to the LeNet model on MNIST handwritten digit dataset.

II. Proposed Approach

Figure 1 gives an overview of the test cases generation method. The specific procedures are introduced as follows.

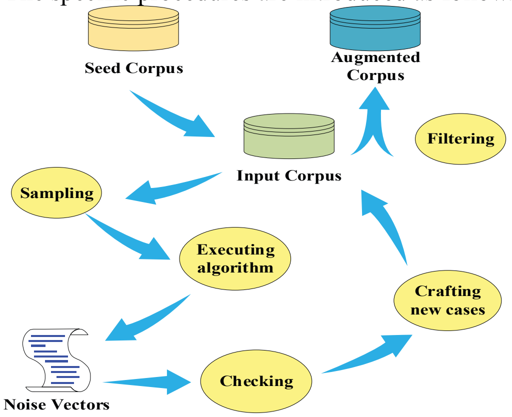

Figure 1. The overall framework of the method.

Sampling Procedure: The sampling strategies are usually heuristic. In our algorithm, the input corpus is sampled according to the input domain data distribution. For example, the MNIST dataset has ten categories, and the numbers of samples in every categories are equal. So we take the same number of samples from each category randomly. This sampling method can ensure that the generated new test

[*] Corresponding author.

978-1-7281-3925-8/19/$31.00 ©2019 IEEE
DOI 10.1109/QRS-C.2019.00104

set roughly obeys the data distribution of the input domain, and the quality of the test set is guaranteed to some extent.

Executing Algorithm procedure: We apply a simple adverisal sample generation algorithm, DeepFool[2], to get the noise vectors. The obtained noise vectors are always too small to cause great changes for the behavior of the model under test. On the other hand if the magnitude of the noise is too tiny, within certain test time, the generated test cases cannot approach the classification boundary at all, which deviates from our original intention. In order to solve this problem, we introduce a "step size" parameter n to control the executing frequency of the DeepFool algorithm.

Checking procedure: The checking procedure determines whether the noise vector is reserved or not. In our method, we use the deformation degree from the seed use case to the new test case as the inspection standard. A study in [3] indicates that as long as the deformation degree is less than 14.29%, the human eyes can still correctly identify the original images and correctly classify them. We apply this metric to our algorithm.

Filtering procedure: We have applied a filtering procedure to the framework. The test cases are screened out based on a time-series attribute. While maintaining the input corpus, we use the attribute to indicate how many times mutation have been made in each test case. Initially, all seed use cases in the input corpus have attribute values of 0. When a new test case is added to the input corpus, the attribute value equals its original test case's attribute value plus 1. The filtering strategy is to randomly extract each of the categories proportsdadsasssadsssionally according to magnitude of attribute values.

III. Experiments

We implement the proposed method and perform experiments on classic DNNs and datasets.

We choose a widely used dataset, MNIST, as the experimental dataset. The MNIST dataset is a handwritten digital dataset, including 60,000 training data and 10,000 test data. Each piece of data in MNIST is a grayscale image of size 28*28*1. We used three classic LeNet family DNN models for analysis. We trained the DNN models for 100 epochs and the performance is shown in the following table. In order to evaluate the quality of the generated test set, we use the coverage metric based on the states of neural network.

TABLE I. DNN Models and Their Training Performance

Model	*Neuron*	*Layer*	*Train Loss*	*Train Acc.*	*Test Acc.*
LeNet-1	52	7	0.125	0.987	0.985
LeNet-4	148	8	0.065	0.995	0.994
LeNet-5	268	9	0.063	0.997	0.995

In the 10000 test samples of the MNIST dataset, we randomly select 200 samples from each category, totaling 2000 samples, as seed corpus and then input them into our implementation algorithm. We set the "step size" parameter of the DeepFool algorithm to n=3,5 and 10. During the final procedure, we screen out 4000 test cases, 20% of which come from test cases with time-series attribute less than the median value, and 80% come from test cases greater than the median value.

IV. Preliminary Results and Conclusions

We call the test case a good test case when it makes DNN produce a new coverage. It can be seen from the data in the table that lots of good test cases have been generated. So our method can generate high-quality test sets to a certain extent, and can effectively expand the test case sets when the test data is insufficient.

TABLE II. NGTC for Original Set and Mutated Set

DNN Model	*Step Size*	*NGTC[a] for Original Set*	*NGTC for Mutated Set*
LeNet-1	n=3	926	1924
	n=5	926	1943
	n=10	926	1928
LeNet-4	n=3	861	1719
	n=5	861	1757
	n=10	861	1750
LeNet-5	n=3	781	1455
	n=5	781	1502
	n=10	781	1486

a. NGTC refers to number of good test cases.

We have proposed a test cases generation technique based on an adversarial samples generation algorithm for image classification DNNs, which can well solve the problem of the augmentation of the test set. We have also implemented framework of the algorithm, and have carried out a simple experiment on the MNIST dataset and LeNet models. The experimental results show that the method is effective. The current research on testing of DL software seems to have entered the bottleneck. On the one hand, the theoretical basis of the proposed methods cannot be proved and explained. On the other hand, the practicality of the proposed algorithms is relatively low. Based on an adversarial samples generation algorithm, we have proposed a feasible method for generating test cases. This method also has its own defects, but it can be combined with the CGF methods. Both our methods and the CGF methods make pixel-level modifications to the images, and our future work will consider the possibility of combining our method with image transformation methods.

Acknowledgment

This work is supported by National Key R&D Program of China(No:2018YFB1403400).

References

[1] Odena, Augustus and Ian J. Goodfellow. "TensorFuzz: Debugging Neural Networks with Coverage-Guided Fuzzing." CoRR abs/1807.10875 (2018): n. pag.

[2] Moosavi-Dezfooli, Seyed Mohsen , A. Fawzi , and P. Frossard . "DeepFool: a simple and accurate method to fool deep neural networks."(2015).

[3] Papernot, Nicolas , et al. "The Limitations of Deep Learning in Adversarial Settings." (2015).