Behavior Pattern-Driven Test Case Selection for Deep Neural Networks

Yanshan Chen[1] Ziyuan Wang[1*] Dong Wang[2] Yongming Yao[3] Zhenyu Chen[2]
[1] School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China
[2] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[3] Tongda College, Nanjing University of Posts and Telecommunications, Yangzhou, China
*Corresponding: wangziyuan@njupt.edu.cn

***Abstract*—With the widespread application of deep learning systems, the robustness of deep neural networks (DNNs) is received increasing attentions recently. By studying the distribution of neurons outputs in DNN models, we found that the behavior patterns of neurons are different for different kinds of DNNs' inputs, e.g. test cases generated by different adversarial attack techniques. In this paper, we extract the neuron behavior patterns of DNNs under different adversarial attack techniques, use them as the guidance for test case selection. Experimental results show that this method is more efficient than random technology.**

***Keywords*—deep neural network, behavior pattern, test case selection, test case prioritization**

I. Introduction

Artificial intelligence has been widely utilized in applications. Especially, deep learning (DL) has become popular since its high accuracy. During the wide-scale deployment of deep neural networks (DNNs) in those DL systems, many types of attacks, e.g. adversarial examples, have been reported. Software testing techniques could help to assure the quality of DL systems by detecting vulnerabilities of DNNs at an early stage. Many people paid their attention on the coverage-based testing techniques and proposed some white-box testing criteria for DNNs.

As we known, a deep neural network consists of neurons in multiply layers and connections among neurons in neighbor layers, where each neuron is a computational unit that computes the output through an activation function. Traditional coverage-based testing techniques, e.g. *DeepXplore* [1] and *DeepGauge* [2], attempted to increase the neuron-level coverage percentage, which reflects the diversity of neurons output behaviors, to guarantee the testing adequate. Different test cases will cause different neurons outputs, and different groups of test cases will cause different patterns of neurons output behaviors (short for *behavior pattern*).

In this paper, based on the neurons outputs, we define neuron behavior for test case of DNN model and define neuron behavior pattern for test set of DNN model. By examining some datasets and models, we found that the behavior patterns of neurons are different for different kinds of DNNs' inputs, e.g. test cases generated by different adversarial attack techniques. For each group of test cases that generated by the same adversarial attack technique, there is a convergent behavior pattern. It means that we could extract behavior pattern from only partial test cases in that group. Such a behavior pattern may help people to testing DL systems effectively.

We focus on behavior pattern-driven test case selection problem for deep neural networks in this paper. The neuron behavior patterns under different adversarial attack techniques are extracted and utilized as the guidance for test case prioritization. Experimental results show that prioritized test cases could hit adversarial examples more rapidly.

II. Behavior Pattern

Suppose there is a DNN model with a set of neurons $N = \{n_1, n_2, ..., n_k\}$ and a test set $T = \{x_1, x_2, ..., x_m\}$. For a given $n_i \in N$ and a given $x_j \in T$, the neurons output value obtained from the activation function is $out(n_i, x_j)$. For simplicity, we use a Boolean state to replace the float value:

$$a(n_i, x_j) = \begin{cases} 1, & \text{if } out(n_i, x_j) > t \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Where t is a threshold to determine the activation state. The neuron behavior for a test case $x_j \in T$ is a vector:

$$B(x_i) = [a(n_1, x_i), a(n_2.x_i), ..., a(n_k.x_i)] \tag{2}$$

Where there are k Boolean states of k neurons.

For the test set $T = \{x_1, x_2, ..., x_m\}$, the behavior pattern could be defined as a mean vector:

$$BP(T) = \frac{\sum_{x_i \in T} B(x_i)}{|T|} = [bp_1(T), bp_2(T), ..., bp_k(T)] \tag{3}$$

For each group of test cases that generated by an adversarial attack technique, there is a convergent behavior pattern, which could be extracted from partial test cases in that group.

For 3 DNN models on MNIST dataset (see section 3 in detail), the difference between behavior pattern of passed test set (right inputs) and behavior patterns of failed test sets (adversarial examples) generated by each adversarial attack technique could be found in Table 1. The difference between two test set T_1 and T_2 is measured by a L1-norm distance:

$$dist = \sum_{i=i}^{k} |bp_i(T_1) - bp_i(T_2)| \tag{4}$$

From the data in the table we can see the behavior patterns under different kinds of inputs are different.

978-1-7281-0492-8/19/$31.00 ©2019 IEEE
DOI 10.1109/AITest.2019.000-2

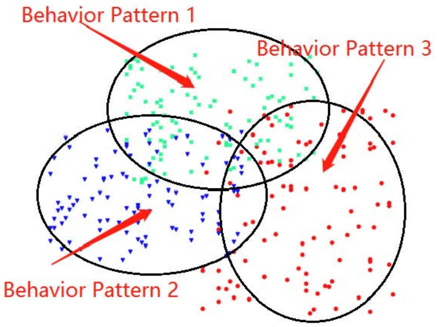


Fig. 1: Behavior patterns of different groups of test cases

TABLE I: Difference between behavior pattern of passed test set and behavior patterns of failed test sets

	FGSM	JSMP	Gaussian	Uniform
right	2.736 3.388 15.685	1.563 2.106 10.452	7.827 7.744 21.019	7.713 8.116 21.751
FGSM		2.955 2.719 9.649	6.849 6.239 14.673	6.702 6.538 14.97
JSMP			7.814 7.181 16.529	7.705 7.564 17.333
Gaussian				0.533 0.776 1.5039

III. Test Case Prioritization

There are significant differences between behavior patterns from different kinds of test cases. Such a behavior patterns difference could help people to select test cases to detect errors in DNN model efficiently. In this paper, we attempt to utilize behavior pattern in test case prioritization.

Approach. Additional greedy strategy is adopted in our approach. In each step, a test case with the greatest metric values is selected from the test case pool that contain both passed test cases (right inputs) and failed test cases (adversarial examples). Such a process stops until all test cases have been selected into a prioritized test suite or meet some other termination conditions.

In order to select adversarial examples as rapidly as possible, the metric value for test selection is the similarity between the behavior of test case under evaluate and the behavior pattern of a given type of adversarial examples:

$$similarity(x) = \sum_{n \in N}(w(x, n_i)) \tag{5}$$

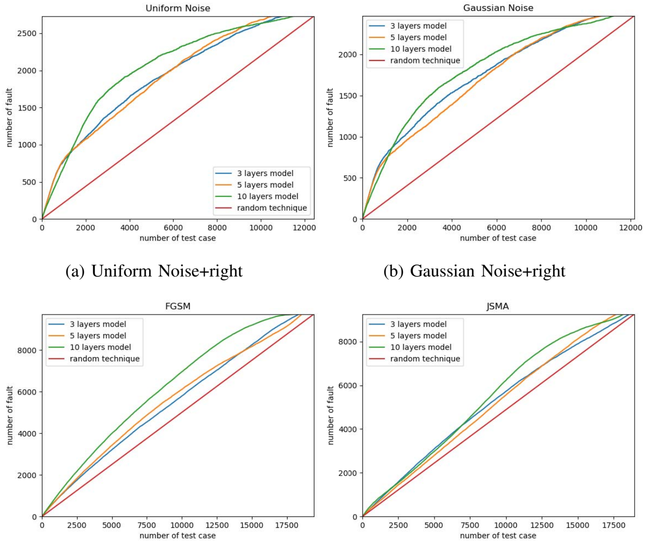
Where each neuron n_i contributes its own weight $w(x, n)$ for $similarity(x)$:

$$w(n_i) = \begin{cases} 1, & |a(x, n_i) - bp_i(TA)| < |a(x, n_i) - bp_i(TR)| \\ -1, & \text{otherwise} \end{cases} \tag{6}$$

Where TA is a small set of adversarial examples generated by a given adversarial attack technique and TR is a small set of right inputs.

Experiment. We chose MNIST, a popular public data set for handwritten digital recognition as our experimental datas. We select 3 full-connected deep neural networks models, which include 3, 5 and 10 hidden layers respectively. We use 4 mature adversarial attack techniques, including FGSM, JSMA, and two decision-based techniques Gaussian Noise and Uniform Noise, to generate adversarial examples. We mix all the right inputs with all the adversarial examples generated by one of four adversarial attack techniques to form 4 mixed test case pools that contain both right inputs and adversarial examples. The threshold, which determine whether the neuron output is activated or not, is set as 0.

The speed of adversarial example detection for 4 mixed test case pools are displayed in 4 sub-figures in Fig. 2. The red lines denote the theory speed of random technique, and other three lines in each sub-figure denote fault detection speed of prioritized test suite generated for three DNN models.



(a) Uniform Noise+right

(b) Gaussian Noise+right

(c) FGSM+right

(b) JSMA+right

Fig. 2: Test Case Selection Result

Remarks. Experimental results show that prioritized test cases could detect faults more rapidly than random technique. It means that the behavior pattern-driven test case selection technique could select adversarial examples from the mixed test case pool that contain both passed and failed test cases.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (61832009, 61772259).

References

[1] Pei K, Cao Y, Yang J, et al. Deepxplore: Automated whitebox testing of deep learning systems. Proceedings of the 26th Symposium on Operating Systems Principles(SOSP). ACM, 2017: 1-18.

[2] Ma L, Juefei-Xu F, Zhang F, et al. Deepgauge: Multi-granularity testing criteria for deep learning systems. Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering(ASE). ACM, 2018: 120-131.

[3] Brendel W, Rauber J, Bethge M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. International Conference on Learning Representations (ICLR), 2018.