Towards Reliability Assurance of Deep Learning Systems

Jianjun Zhao
Kyushu University

CSBSE 2018, November 17, 2018

1

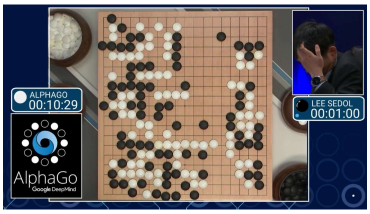
Pangu Research Group
(https://pangukaitian.github.io/pangu/en/index.html)

- Our group focuses on researches on the potential symbioses between software engineering (SE) and artificial intelligence (AI)
- The overall goal is to obtain better software and AI systems, making them more robust, reliable, and secure, and easier to specify, build, maintain, or improve
- Our group consists of 2+ faculty members, 3 PhD students, and 7 master students

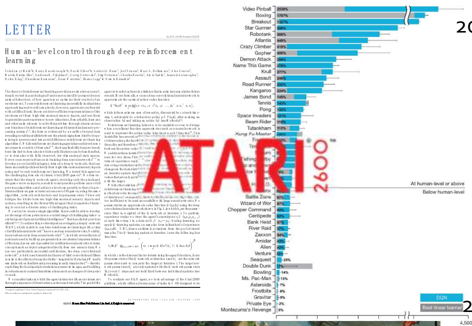
2

Deep Learning Matches Human Intelligence

AlphaGo 4:1 Human Champion 2016



AlphaGo ZERO 100:0 AlphaGo 2017



2015

2017

3

Rushing for Development and Real-world Deployment

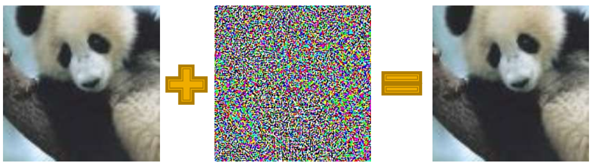
....

4

Driving Force of Many Novel Tech.

5

Current Deep Learning is Vulnerable

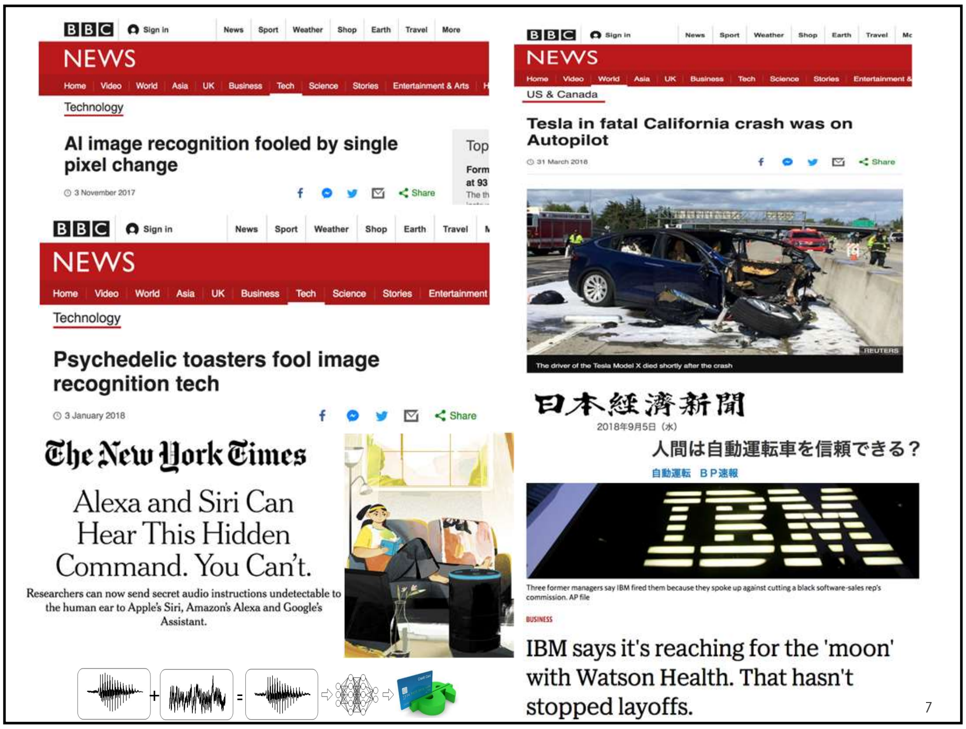


Classified as panda | Small adversarial noise | Classified as gibbon

Ian Goodfellow, Jon Shlens, Christian Szegedy, Explaining and Harnessing Adversarial Examples, ICLR, 2014

Accident

6



Reliability Assurance for DL is at Early Stage

DeepXplore SOSP'17	DeepTest ICSE'18
ML Fairness Testing ASE'18	DeepRoad ASE'18
DeepConcolic ASE'18	DeepGauge ASE'18
TensorFlow Program Bugs ISSTA'18	MODE: DNN Debugging FSE'18

……

8

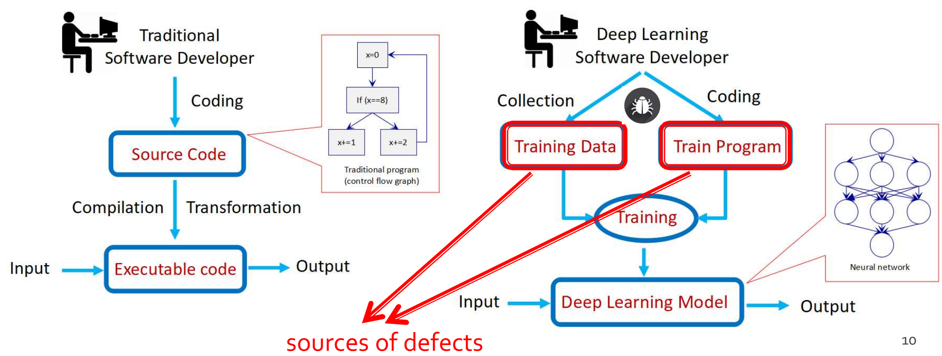
Reliability Assurance for Traditional Software Testing Criteria and Tools

- Line Coverage
- Branch Coverage
- Function Coverage
- Data Flow Coverage
- Combinatorial Coverage
- Mutation testing Coverage
- ...

9

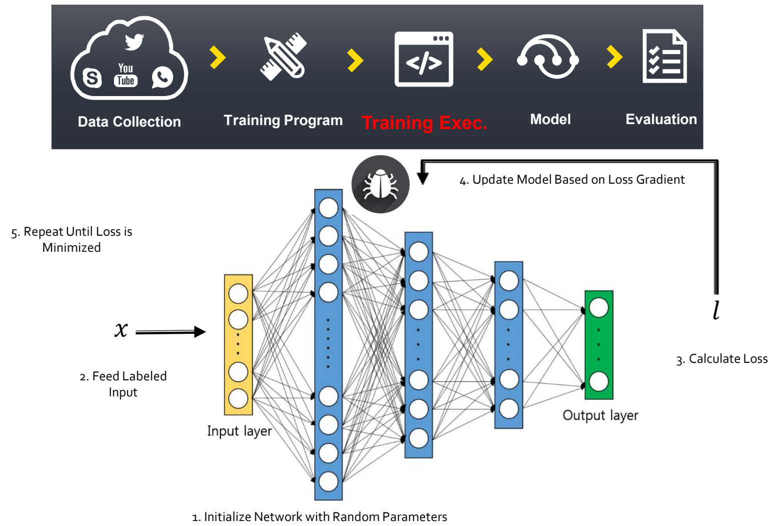
Source of Defects Programming Paradigm

- The decision logic of a traditional software:
  - In the form of code
- The decision logic of a DL system:
  - The structure of DNN
  - The connection weights



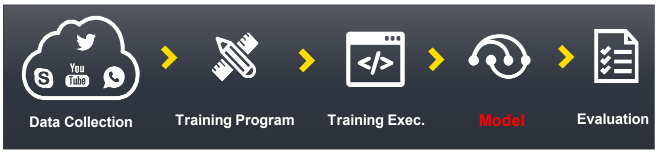
10

Source of Defects Training Procedure

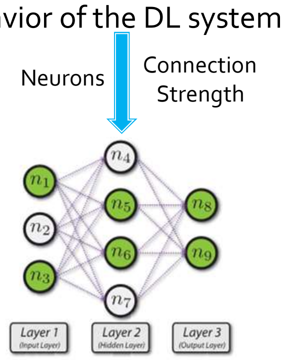


11

Lacking of Interpretability and Understandability

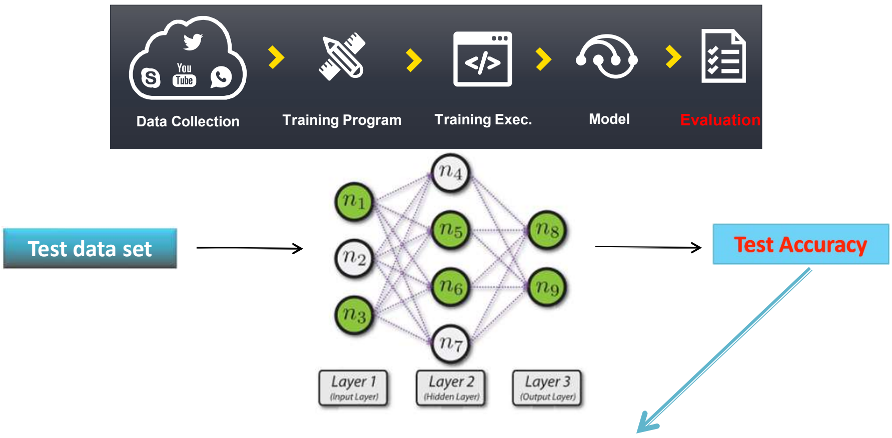


Behavior of the DL system ?



12

Reliability Measurement Immature



High **accuracy** $\not\rightarrow$ High **DL quality**

13

Towards Reliability Assurance for Deep Learning Software

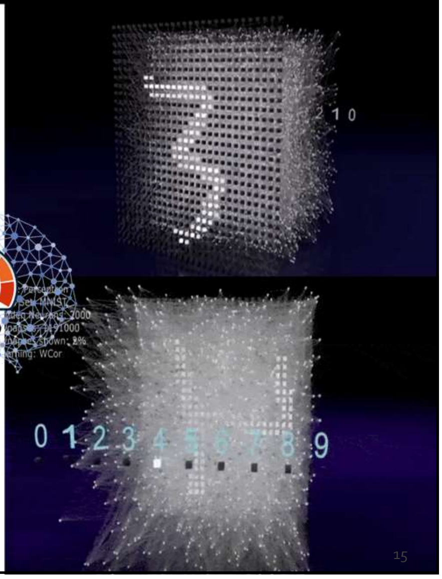
- **Testing quality & logic coverage** — DeepGauge
- **MT-based test data quality evaluation** — DeepMutation
- **Combinatorial latent space exploration** — DeepCT

14

Overview of DeepGauge (ASE2018)

Multi-Granularity Testing Criteria for Deep Learning Systems

- **Enable** quality evaluation of DLs from multiple portrayals
- **Provide** systematic guidance of TestGen for Defects
- **Facilitate** Interpretation & Understanding



DeepGauge

15

Design of DeepGauge

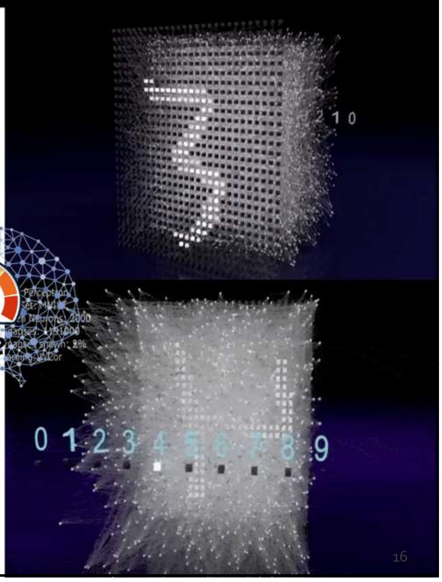
Simple to understand & use — Simplicity

Efficient to compute — Efficiency

General to diverse DNNs — Generality

Scale to large DNNs — Scalability

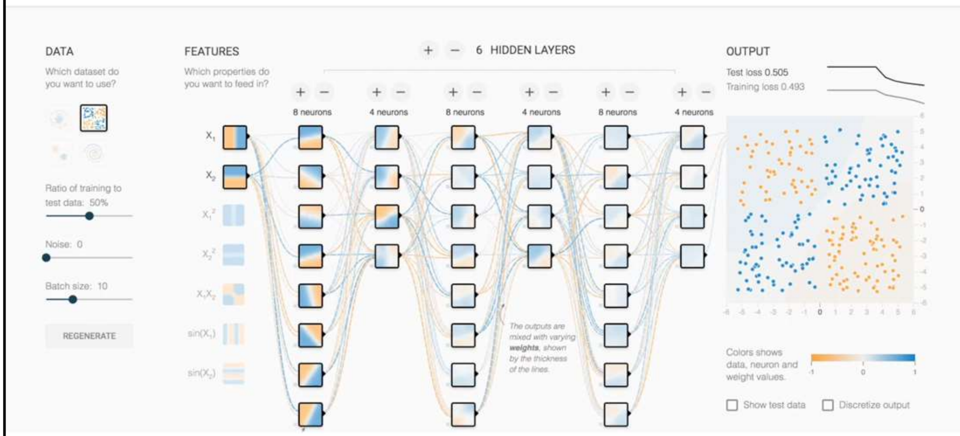
Adaptable by cases — Adaptability



DeepGauge

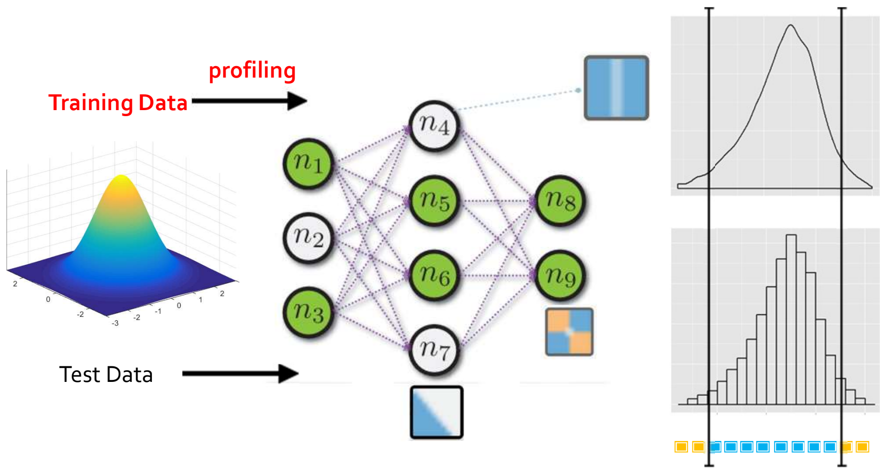
16

The Role of Neurons and Layers
(from TensorFlow Neural Network Playground)



17

Neuron Level Criteria



18

Neuron Level Criteria

Large Scale Test Data —Neurons→

K-Multisection Neuron Coverage

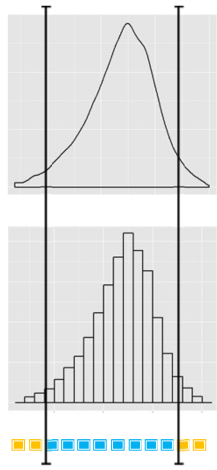
$$\mathrm{KMNCov}(T,k) = \frac{\sum_{n\in N} |\{S_i^n \mid \exists \mathbf{x} \in T: \ \phi(\mathbf{x},n) \in S_i^n\}|}{k \times |N|}$$

Neuron Boundary Coverage

$$\mathrm{NBCov}(T) = \frac{|\mathrm{UpperCornerNeuron}| + |\mathrm{LowerCornerNeuron}|}{2 \times |N|}$$

Strong Neuron Activation Coverage

$$\mathrm{SNACov}(T) = \frac{|\mathrm{UpperCornerNeuron}|}{|N|}$$



19

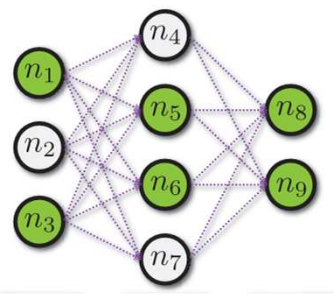
Layer Level Criteria

Large Scale Test Data —Layers→

Top-K Neuron Coverage

$$\mathrm{TKNCov}(T,k) = \frac{|\bigcup_{\mathbf{x}\in T}(\bigcup_{1\le i\le l} \mathrm{top}_k(\mathbf{x},i))|}{|N|}$$

Neuron interactions of same layer



Top-K Neuron Patterns

$$\mathrm{TKNPat}(T,k) = |\{(\mathrm{top}_k(\mathbf{x},1),\ldots,\mathrm{top}_k(\mathbf{x},l)) \mid \mathbf{x} \in T\}|$$

Neuron interactions across layers

20

Large Scale Empirical Study

MNIST
60,000
10,000
(28,28,1)
784 dim.

IMAGENET
(LSVRC-2012)
1,000,000+
50,000
(224,224,3)
150528 dim.

DNNs	#Neurons	#Layers
LeNet-1	52	7
LeNet-4	148	8
LeNet-5	268	9
VGG-19	16,168	25
ResNet-50	95,059	176

21

Large Scale Empirical Study

Adversarial Test Gen
FGSM
BIM
JSMA
C&W

Coverage Eval.
KMNC
NBC
SNAC
TKNC
TKNP
NC

DNNs
Test org.

Coverage Parameters

Coverage Criteria	Parameter setting		
KMNC	k=1,000	k=10,000	N.A
NBC	LB=l	LB=l-0.5*σ	LB=l-σ
	UB=u	UB=u+0.5*σ	UB=u+σ
SNAC	UB=u	UB=u+0.5*σ	UB=u+σ
TKNC	k=1	k=2	k=3
TKNP	k=1	k=2	k=3
NC	th.= 0.0, 0.25, 0.5, 0.75		

5 DNN Profiling

18 Group Large Scale Adv Test Gen

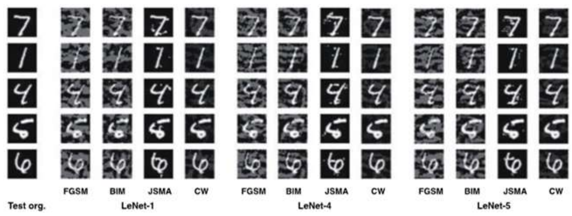
190,000 Test data= 150,000 (MNIST) + 40,000 (ImageNet)

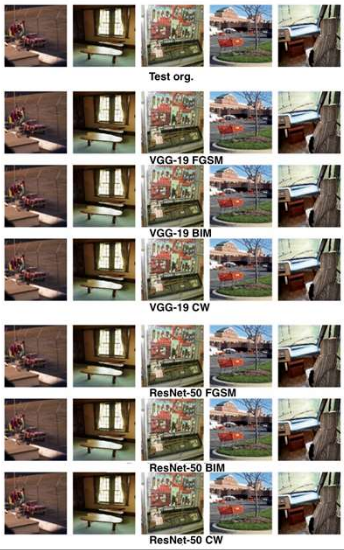
414 Evaluation Configurations= 270 (MNIST) + 144 (ImageNet)

NVIDIA **Tesla M40 GPU 24GB /** 18-core 2.3GHz Xeon 64-bit 196 GB

22

Adversarial Examples





- CW and JSMA make smaller perturbation
- Adv on ImageNet is often imperceptible
- JSMA and CW is computation intensive
- Even more obvious on ImageNet subjects

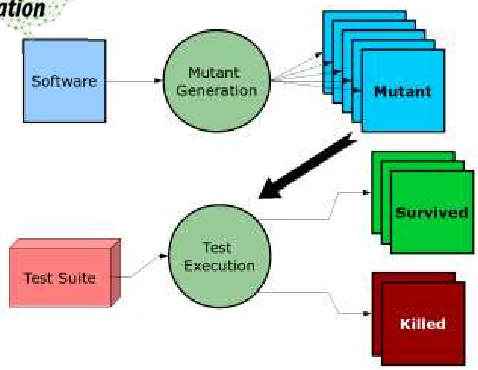
DeepMutation: Mutation Testing of Deep Learning System (ISSRE'18)


DeepMutation: Mutation Testing of
Deep Learning Systems

arXiv:1805.05206v2 [cs.SE] 14 Aug 2018

Testdata Quality Assessment

Through Mutation Testing



24

Combinatorial Testing

DeepCT

arXiv:1806.07723v1 [cs.SE] 20 Jun 2018

Combinatorial Testing for Deep Learning Systems

Lei Ma[1,2], Fuyuan Zhang[2], Minhui Xue[3], Bo Li[4], Yang Liu[2], Jianjun Zhao[5], and Yadong Wang[1]

[1] Harbin Institute of Technology, China
[2] Nanyang Technological University, Singapore
[3] New York University Shanghai, China
[4] University of Illinois at Urbana–Champaign, USA
[5] Kyushu University, Japan
Contact author: malei@hit.edu.cn

Abstract. Deep learning (DL) has achieved remarkable progress over the past decade and been widely applied to many safety-critical applications. However, the robustness of DL systems recently receives great concerns, such as adversarial examples against computer vision systems, which could potentially result in severe consequences. Adopting testing techniques could help to evaluate the robustness of a DL system and therefore detect vulnerabilities at an early stage. The main challenge of testing such systems is that its runtime state space is too large: if we view each neuron as a runtime state for DL, then a DL system often contains massive states, rendering testing each state almost impossible. For traditional software, combinatorial testing (CT) is an effective testing technique to reduce the testing space while obtaining relatively high defect detection abilities. In this paper, we perform an exploratory study of CT on DL systems. We adapt the concept in CT and propose a set of coverage criteria for DL systems, as well as a CT coverage guided test generation technique. Our evaluation demonstrates that CT provides a promising avenue for testing DL systems. We further pose several open questions and interesting directions for combinatorial testing of DL systems.

Keywords: Combinatorial testing · Deep learning · Adversarial attacks

1 Introduction

Deep learning (DL) systems have been widely applied in various applications due to their high accuracy, such as computer vision [35], natural language processing [39], autonomous driving [5], and automated medical diagnosis [33]. However, recently DL systems have been shown to be vulnerable against different attacks, such as adversarial examples in computer vision and audio systems. Given that more and more safety-sensitive applications start to adopt DL, deploying DL without thorough testing to safety-critical applications can lead to severe consequences, such as possible accidents in autonomous driving [12]. DL systems are

- Neuron Activation Configuration
- T-way combination sparse coverage
- T-way combination dense coverage
- (p,t)-completeness coverage

Parameters

P1 : A , B , C
P2 : 1 , 2
P3 : X , Y

All Combinations

		P1	P2	P3
TC1	:	A	1	X
TC2	:	A	1	Y
TC3	:	A	2	X
TC4	:	A	2	Y
TC5	:	B	1	X
TC6	:	B	1	Y
TC7	:	B	2	X
TC8	:	B	2	Y
TC9	:	C	1	X
TC10	:	C	1	Y
TC11	:	C	2	X
TC12	:	C	2	Y

2-Pair (Pairwise)

		P1	P2	P3
TC1	:	A	1	X
TC4	:	A	2	Y
TC6	:	B	1	Y
TC7	:	B	2	X
TC9	:	C	1	X
TC12	:	C	2	Y

25

NVIDIA AI Tech Center (NVAITC)

10+ Nvidia Tesla M40, 20 Tesla V100

DeepGauge DeepMutation DeepCT DeepLaser

http://deepgauge.github.io/