Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

深度学习系统的自动化测试简介

深度神经网络（DNN）自动化测试

硕士研究生　王逸洲

2020年07月12日

内容提要

- 背景简介
- 基本概念
- 算法原理
- 优劣分析
- 应用总结
- 参考文献

背景简介

- 预期收获
  - 理解深度学习系统测试与传统软件测试的区别
  - 理解基于覆盖的深度学习系统测试方法原理
  - 了解深度学习系统测试的应用前景

背景简介

- 深度学习（DL）在图像分类、语音识别等领域达到或超过了人类水平的性能，且被广泛应用于安全关键领域中（自动驾驶、恶意软件检测等）。

- 一些原因（如训练数据偏差、模型过拟合或欠拟合），会导致深度学习系统在一些边角案例（corner cases）中表现出意料之外或错误的行为。

背景简介

- 当前DNN系统测试的标准方法：
  - 收集并手动标记尽可能多的真实世界测试数据
  - 通过模拟生成合成训练数据（例如Google自动驾驶汽车）

- 存在问题：
  - 手动标记大量复杂、高维度数据需要的人力成本高
  - 测试覆盖低，现有的DNN测试方案都没有尝试覆盖DNN的不同规则，测试输入常常无法揭示DNN的不同错误行为

基本概念

- 基本概念
  - DNN系统：指包含至少一个深度神经网络（DNN）组件的软件系统。
  - 模糊测试：是一种软件测试技术，其核心思想是自动或半自动的生成随机数据输入到一个程序中，并监视程序异常。
    - 基本步骤：
      - 识别目标系统
      - 确定输入
      - 生成模糊数据
      - 使用模糊数据执行测试
      - 监控系统行为
      - 记录缺陷

基本概念

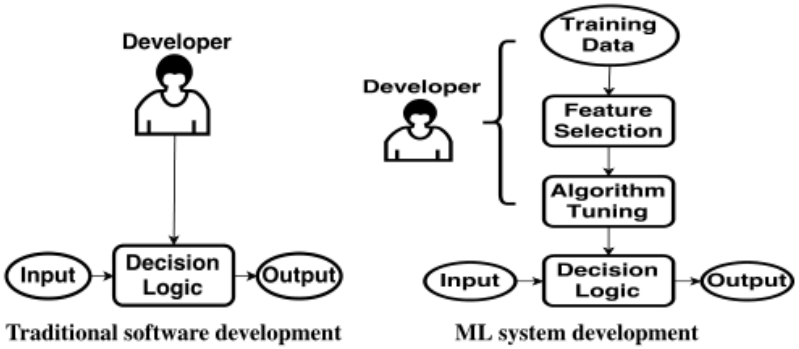
- 传统软件白盒测试的覆盖标准
  - 函数覆盖率
  - 指令覆盖率
  - 判断覆盖率
  - 条件覆盖率
  - 条件/判断覆盖率
  - …

- 深度神经网络系统的测试标准
  - ？？？

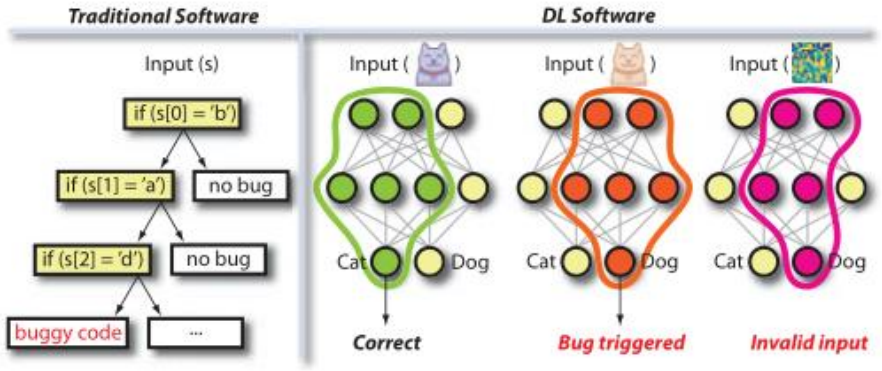
传统软件的测试标准能否直接用于DNN系统的测试?

基本概念

- 传统软件系统开发与深度学习（机器学习）系统的区别



- 传统软件系统开发
  - 决策逻辑由开发人员手动编写
  - 根据不同的条件、指令、判断等产生不同输出
  - 决策逻辑是白盒



- 深度学习系统的开发部署
  - 决策逻辑通过模型训练自动生成
  - 根据每层神经元的输出等产生不同输出
  - 决策逻辑是黑盒

算法原理——DeepXplore

T	基于种子测试样本，自动生成测试样本完成测试
I	无标签的测试种子样本（Seed）、待测试DNN模型
P	1. 选择种子样本 2. 梯度计算 3. 输入待测试模型 4. 判断是否出错
O	高覆盖率测试样本集

P	1.当前DNN系统测试方法缺少覆盖标准 2.当前测试样本选择（生成）方法难以覆盖模型的所有潜在错误
C	DNN网络结构已知，且神经元输出可读取
D	如何生成高覆盖率的测试样本集（如何提高测试覆盖）
L	SOSP 2017（操作系统顶会）

算法原理

- 自动化测试面临的三个问题
  - 如何客观评价测试过程及结果?
  - 测试框架如何判断被测试模型出现错误?（Test Oracle）
  - 如何自动化生成能够使模型产生错误行为的测试样本?

算法原理

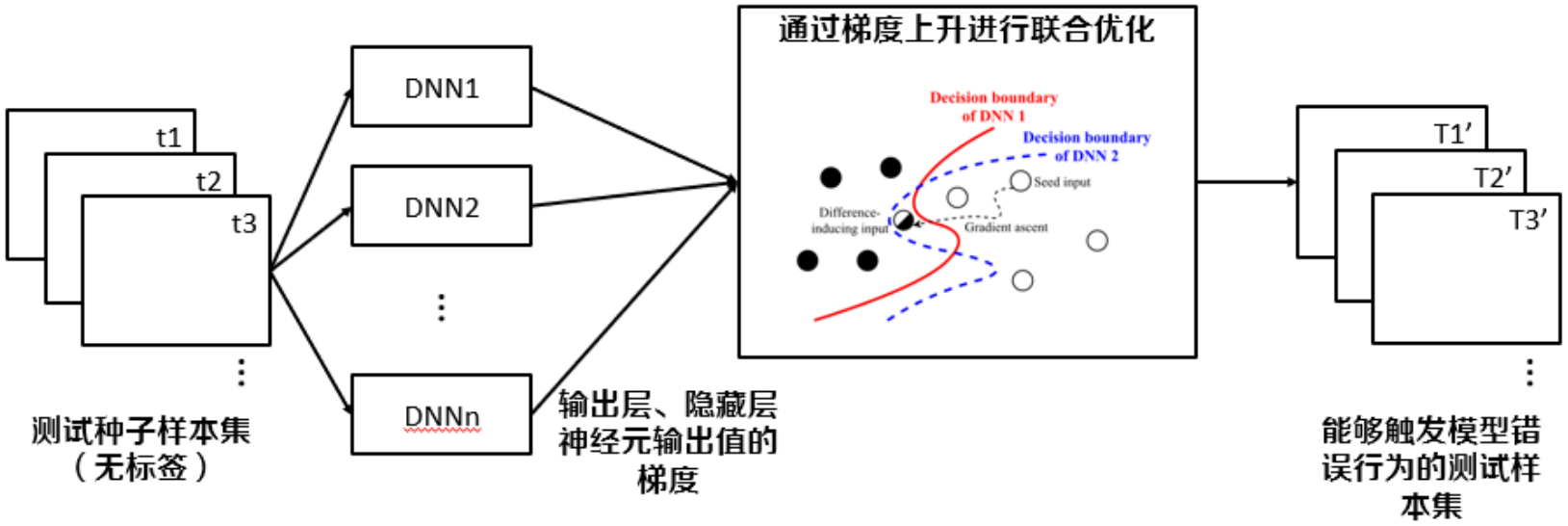
- 如何评价测试结果？（神经元覆盖率）
  - 一组测试输入所激活的神经元数量与DNN网络神经元总数的比值
  - 若神经元的输出值高于设定阈值，则认定神经元被激活

$$NCov(T, \mathrm{x}) = \frac{|\{n|\forall x \in T, out(n,x) > t\}|}{|N|}$$

- $T = \{x_1, x_2, \ldots\}$，表示测试输入样本集
- $N = \{n_1, n_2, \ldots\}$，表示DNN网络的神经元集合
- $out(n,x)$，返回DNN中神经元n对于给定测试输入的输出值的函数

算法原理

- 测试框架如何判断被测试模型出现错误?
- 如何自动化生成能够使模型产生错误行为的测试样本?



- 目标1：生成触发相同任务不同DNN网络差异行为的样本
- 目标2：生成能够最大化神经元覆盖率的测试样本

算法原理

Algorithm 1 Test input generation via joint optimization

Input: **seed_set** ← unlabeled inputs as the seeds
dnns ← multiple DNNs under test
λ_1 ← parameter to balance output differences of DNNs (Equation 2)
λ_2 ← parameter to balance coverage and differential behavior
s ← step size in gradient ascent
t ← threshold for determining if a neuron is activated
p ← desired neuron coverage
cov_tracker ← tracks which neurons have been activated

```
/* main procedure */
gen_test := empty set
for cycle(x ∈ seed_set) do  // infinitely cycling through seed_set
    /* all dnns should classify the seed input to the same class */
    c = dnns[0].predict(x)
    d = randomly select one dnn from dnns
    while True do
        obj1 = COMPUTE_OBJ1(x, d, c, dnns, λ1)
        obj2 = COMPUTE_OBJ2(x, dnns, cov_tracker)
        obj = obj1 + λ2 · obj2
        grad = ∂obj / ∂x
        /*apply domain specific constraints to gradient*/
        grad = DOMAIN_CONSTRNTS(grad)
        x = x + s · grad  //gradient ascent
        if d.predict(x) ≠ (dnns-d).predict(x) then
            /* dnns predict x differently */
            gen_test.add(x)
            update cov_tracker
            break
    if DESIRED_COVERAGE_ACHVD(cov_tracker) then
        return gen_test
```

- $\text{obj}_1(x) = \sum_{k \neq j} F_k(x)[c] - \lambda_1 \cdot F_j(x)[c]$
  - $F_k(x)[c]$:DNN网络k将样本x预测为c的分类概率
  - λ_1：用于平衡DNN网络之间的输出差异
- $\text{obj}_2(x) = f_n(x)$
  - $f_n(x)$表示神经元n对于DNN网络的输入x产生的输出值
- $\text{obj}_{\text{joint}}(x) = \text{obj}_1(x) + \lambda_2 \cdot \text{obj}_2(x)$
  - λ_2：用于平衡覆盖和差异行为的参数

算法原理

- 实验数据集和待测试模型

Dataset	Dataset Description	DNN Description	DNN Name	# of Neurons	Architecture	Reported Acc.	Our Acc.
MNIST	Hand-written digits	LeNet variations	MNI_C1	52	LeNet-1, LeCun et al. [40, 42]	98.3%	98.33%
			MNI_C2	148	LeNet-4, LeCun et al. [40, 42]	98.9%	98.59%
			MNI_C3	268	LeNet-5, LeCun et al. [40, 42]	99.05%	98.96%
Imagenet	General images	State-of-the-art image classifiers from ILSVRC	IMG_C1	14,888	VGG-16, Simonyan et al. [66]	92.6%**	92.6%**
			IMG_C2	16,168	VGG-19, Simonyan et al. [66]	92.7%**	92.7%**
			IMG_C3	94,059	ResNet50, He et al. [31]	96.43%**	96.43%**
Driving	Driving video frames	Nvidia DAVE self-driving systems	DRV_C1	1,560	Dave-orig [8], Bojarski et al. [10]	N/A	99.91%#
			DRV_C2	1,560	Dave-norminit [78]	N/A	99.94%#
			DRV_C3	844	Dave-dropout [18]	N/A	99.96%#
Contagio/Virustotal	PDFs	PDF malware detectors	PDF_C1	402	<200, 200>$^{+}$	98.5%$^{-}$	96.15%
			PDF_C2	602	<200, 200, 200>$^{+}$	98.5%$^{-}$	96.25%
			PDF_C3	802	<200, 200, 200, 200>$^{+}$	98.5%$^{-}$	96.47%
Drebin	Android apps	Android app malware detectors	APP_C1	402	<200, 200>$^{+}$, Grosse et al. [29]	98.92%	98.6%
			APP_C2	102	<50, 50>$^{+}$, Grosse et al. [29]	96.79%	96.82%
			APP_C3	212	<200, 10>$^{+}$, Grosse et al. [29]	92.97%	92.66%

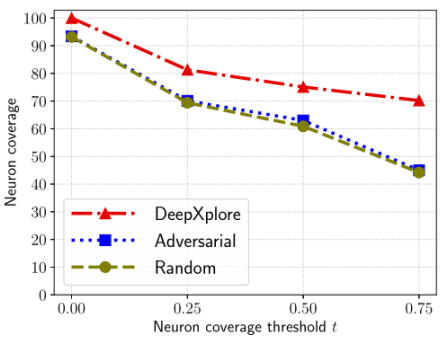
算法原理

- 实验结果

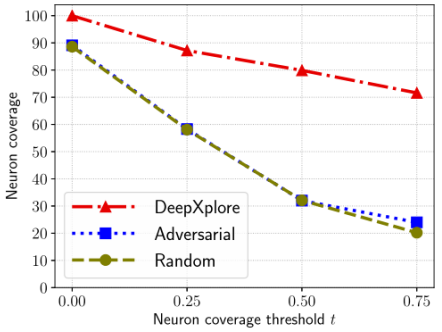
Table 2: Number of difference-inducing inputs found by DeepXplore for each tested DNN obtained by randomly selecting 2,000 seeds from the corresponding test set for each run.

DNN name	Hyperparams (Algorithm 1)				# Differences Found
	λ_1	λ_2	s	t	
MNI_C1	1	0.1	10	0	1,073
MNI_C2					1,968
MNI_C3					827
IMG_C1	1	0.1	10	0	1,969
IMG_C2					1,976
IMG_C3					1,996
DRV_C1	1	0.1	10	0	1,720
DRV_C2					1,866
DRV_C3					1,930
PDF_C1	2	0.1	0.1	0	1,103
PDF_C2					789
PDF_C3					1,253
APP_C1	1	0.5	N/A	0	2,000
APP_C2					2,000
APP_C3					2,000

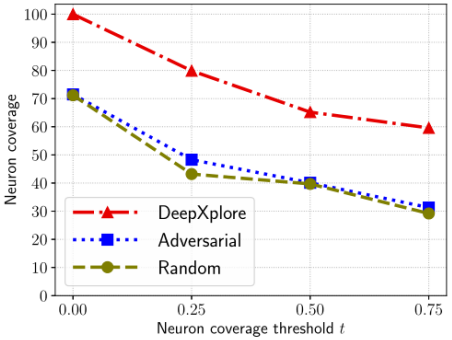
Dataset	Code Coverage			Neuron Coverage		
	C1	*C2*	*C3*	*C1*	*C2*	*C3*
MNIST	100%	100%	100%	32.7%	33.1%	25.7%
ImageNet	100%	100%	100%	1.5%	1.1%	0.3%
Driving	100%	100%	100%	2.5%	3.1%	3.9%
VirusTotal	100%	100%	100%	19.8%	17.3%	17.3%
Drebin	100%	100%	100%	16.8%	10%	28.6%



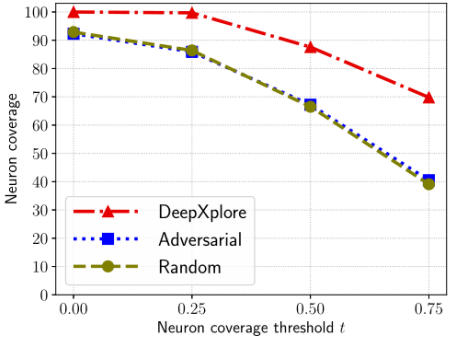
(a) MNIST



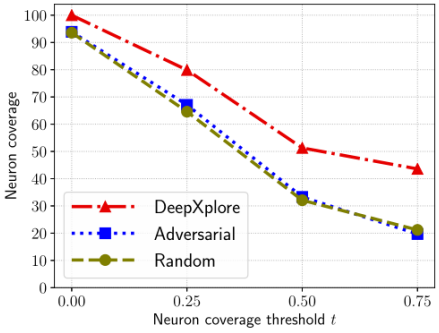
(b) ImageNet



(c) Driving



(d) VirusTotal



(e) Drebin

算法原理

- 神经元覆盖率（NC）的局限性
  - 仅通过NC无法完全表示DNN可能会产生的所有行为
  - 有实验证明通过25个样本即可达到100%覆盖率

- 细粒度测试指标的提出
  - K-段神经元覆盖率：衡量测试样本集对神经元输出区间$[low_n, high_n]$的覆盖范围
  - 神经元边界覆盖率：衡量测试样本集对模型功能边界区域的覆盖程度
  - Top-k神经元覆盖率：衡量测试样本集对模型每一层最活跃的k个神经元的覆盖程度
  - ……

优劣分析

- DeepXplore的优势
  - 填补了深度学习系统测试标准的空白
  - 指导测试样本生成，提高模型、系统的安全性
- DeepXplore的劣势
  - 仅从神经元覆盖率（NC）无法表明测试样本集覆盖了DNN模型的所有可能的输出（25个测试输入可达到100%覆盖）
  - 测试过程依赖多个DNN网络

应用总结

- 应用领域
  - 自动驾驶
  - 恶意软件检测
  - 语音识别
- 未来的发展
  - 循环神经网络（RNN）系统自动化测试方法
  - 复杂结构的深度学习系统的自动化测试
  - 深度学习系统黑盒自动化测试标准的提出

参考文献

[1] Pei K , Cao Y , Yang J , et al. DeepXplore: Automated Whitebox Testing of Deep Learning Systems[J]. 2017.

[2] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chun-yang Chen, Ting Su, Li Li, Yang Liu, Jianjun Zhao, and Yadong Wang. 2018. DeepGauge: Multi-Granularity Testing Criteria for Deep Learning Systems. In Proceedings of the 2018 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE ' 18), September 3 - 7, 2018, Montpellier, France. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3238147. 3238202

[3] https://blog.csdn.net/cindy_cheng/article/details/84323229

知人者智，自知者明。
胜人者有力，自胜者
强。知足者富。强行
者有志。不失其所者
久。死而不亡者，寿。

谢谢！