Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心

面向深度学习软件库的漏洞挖掘方法

硕士研究生 刘力源

2021年12月12日

内容提要

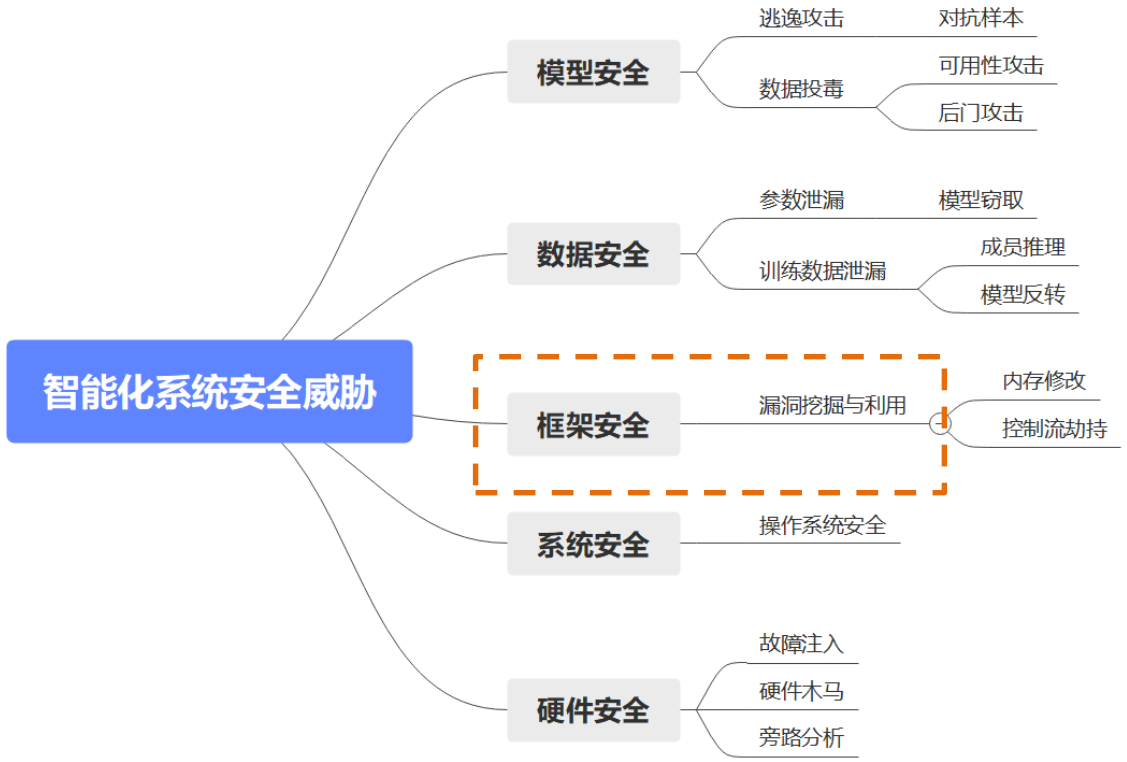
- 背景简介
- 基本概念
- 算法原理
- 优劣分析
- 应用总结
- 参考文献

背景介绍

- 预期收获
  - 1. 了解深度学习软件库的功能及研究其安全性的意义
  - 2. 了解面向深度学习软件库的漏洞挖掘方法
  - 3. 了解现有方法的局限性、改进策略

背景介绍

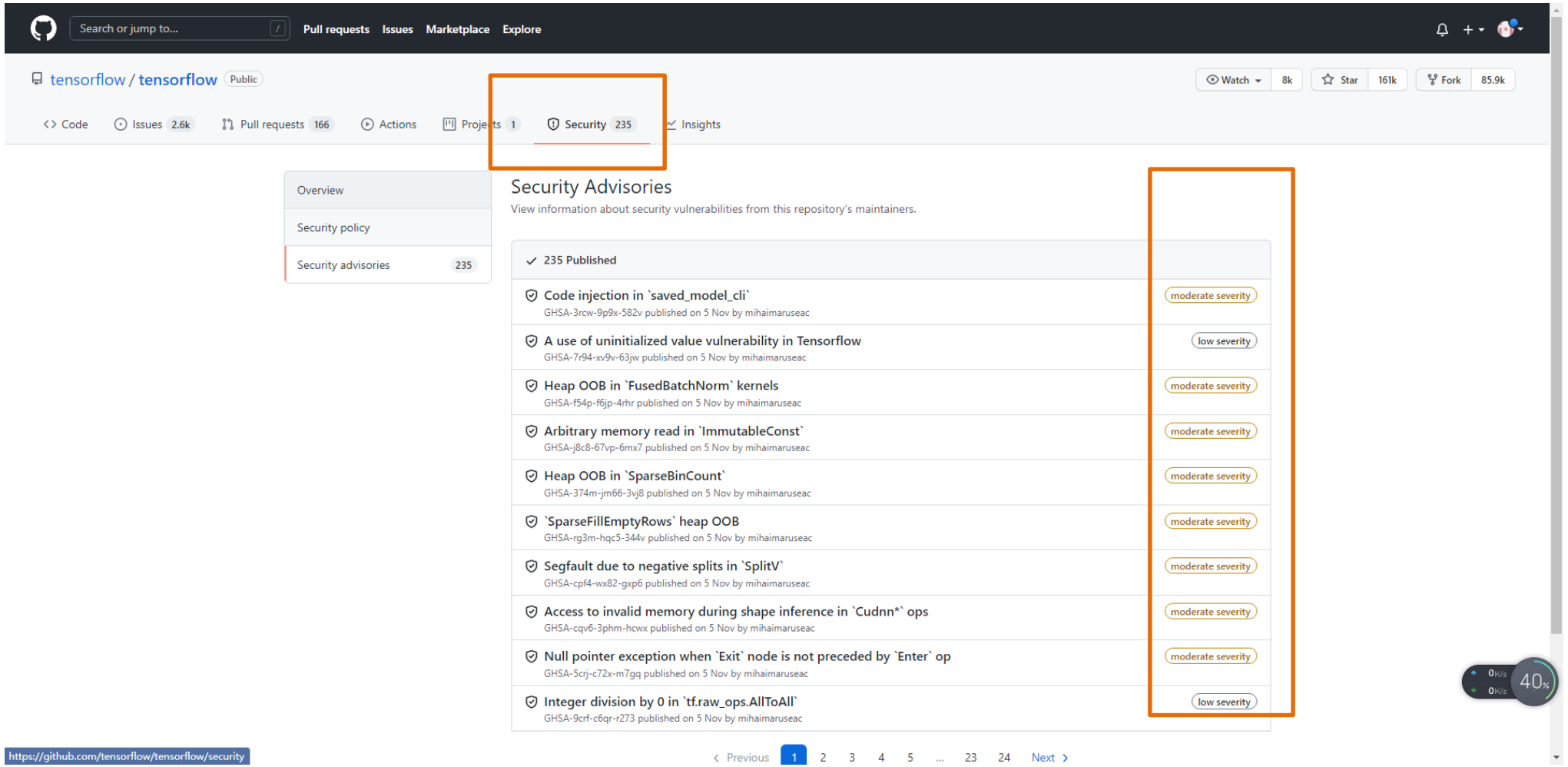
- 研究人工智能安全的最终目标是什么?
  - 功能+性能：预测或分类的实现、准确性
  - 保密、正常运行
- 人工智能系统的安全威胁来自何处?
  - 模型缺陷
  - 外部攻击
  - …
  - 底层组件



- 深度学习算法直接使用较为复杂
  - 解决：开发高阶的应用程序接口
  - 问题：框架带来的安全问题直接影响到上层模型

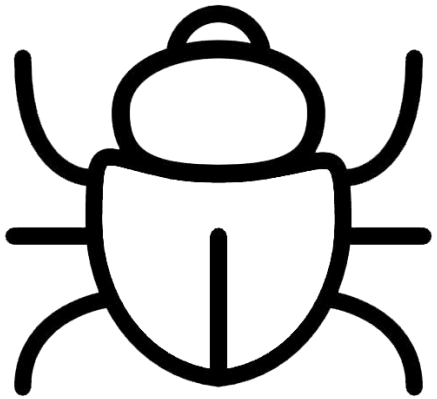
背景简介

- 框架开发者并未重视安全问题



背景简介

- 漏洞
  - 定义：系统设计、实现或操作和管理中的缺陷或弱点，可以被利用来违反系统的安全策略
  - 关于缺陷（Bug）：硬件、软件或协议上的逻辑错误和安全策略弱点
- 漏洞挖掘
  - 静态挖掘
    - 分析程序的词法、语法和语义等，结合数据流、控制流信息
  - 动态挖掘
    - 在实际执行程序的基础上采用的分析技术



基本概念

基本概念

- 深度学习系统典型架构

基本概念

- 深度学习框架/组件/库
  - 框架：集成完整深度学习开发功能的平台
  - 组件&库：各个功能模块（类、函数为主体）
  - 基于Python、C/C++和Java编写
- 作用与优势
  - 模型推理：深度学习核心算法
  - 避免写重复的代码，提高系统开发效率
- 主流深度学习框架
  - Tensorflow、Pytorch、CNTK、Theano、Mxnet…
  - Pandas、Numpy…

基本概念

- 静态挖掘
  - 优势：漏报率低
  - 劣势：依赖于人工经验和专家知识，自动化程度较低
  - 主流方向：运用机器学习或深度学习技术的代码相似性检测

- 动态挖掘
  - 优势：获取程序实际运行过程中的上下文信息，精度较高、误报率较低
  - 劣势（难点）：大量工作放在构建测试输入上
  - 主流方向：模糊测试、符号执行

- 应用局限性
  - 静态：需要专家知识、时间成本高

基本概念

- 动态：构建测试输入，实现软件库的调用
  - 构建API输入
    - 构造能够调用函数、类的输入
    - 局限性：函数输入通常为不同类型数据的组合，包含张量、数据类型、字符串等

```
tf.raw_ops.Conv2D(
    input, filter, strides, padding, use_cudnn_on_gpu=True, explicit_paddings=[],
    data_format='NHWC', dilations=[1, 1, 1, 1], name=None
)
```

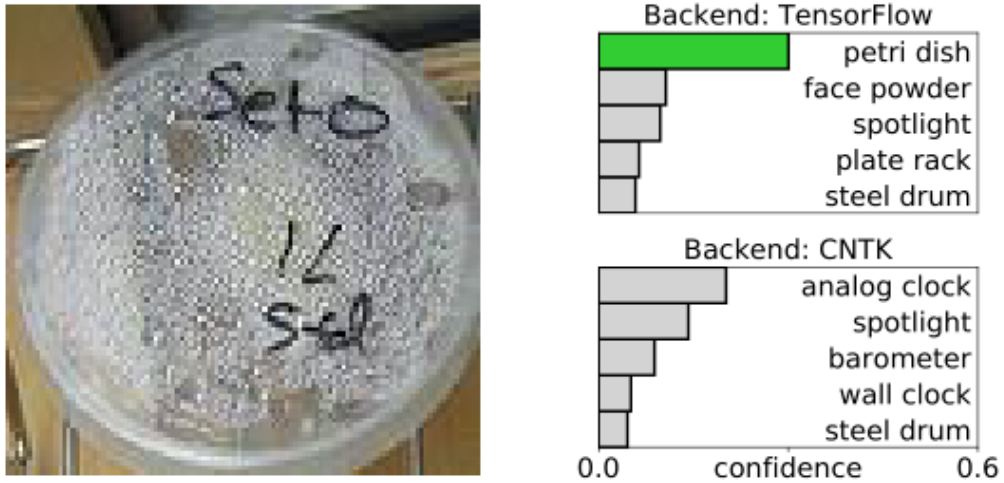
  - 构建模型
    - 在源代码层面基于指定框架运行公开深度学习模型
    - 局限性：极少数的情况会触发崩溃，如何获得测试预期（参照物）

深度学习底层算法往往更为复杂

$$x^4 + x^3 + x^2 + 1 = ?$$

基本概念

- 差分测试
  - 基于框架A和框架B运行同一个深度学习模型，监测异常
    - 崩溃（Crashes）
    - NaN（Not a Number）
    - 输出不一致：置信度 、 分类标签
  - 产生输出不一致的原因
    - 浮点计算精度差异等
    - A or B某一部分存在逻辑错误



(a) Input image “Petri dish” (b) Top-5 InceptiontionResNetV2

算法原理

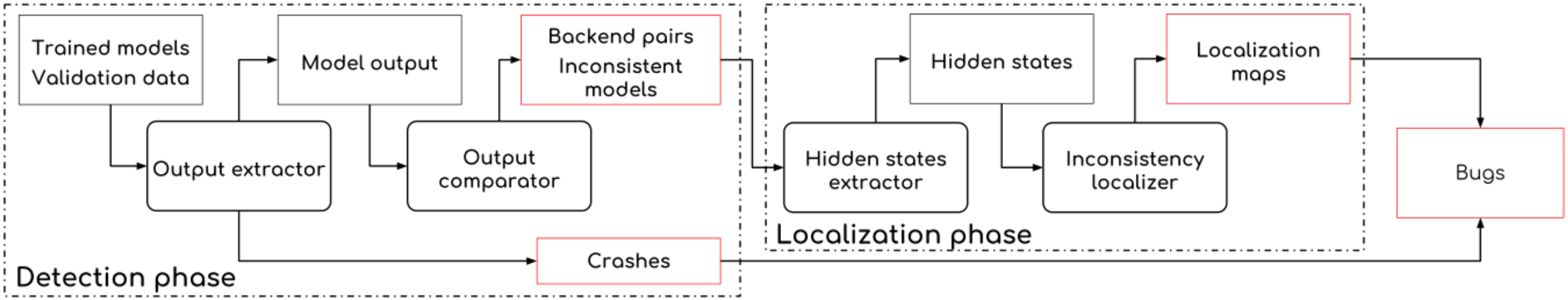
算法原理—CRADLE

T	深度学习组件安全测试
I	框架、模型、数据集
P	1.模型运行 2.输出不一致比较 3.缺陷局部化 4.缺陷分析
O	框架bug及其位置、解释

P	从输出差异中筛选出暴露缺陷的部分；缺陷位置局部化
C	只执行模型测试任务
D	差异度量设定和选取；定位产生差异值的层
L	2019 IEEE/ACM ICSE A类

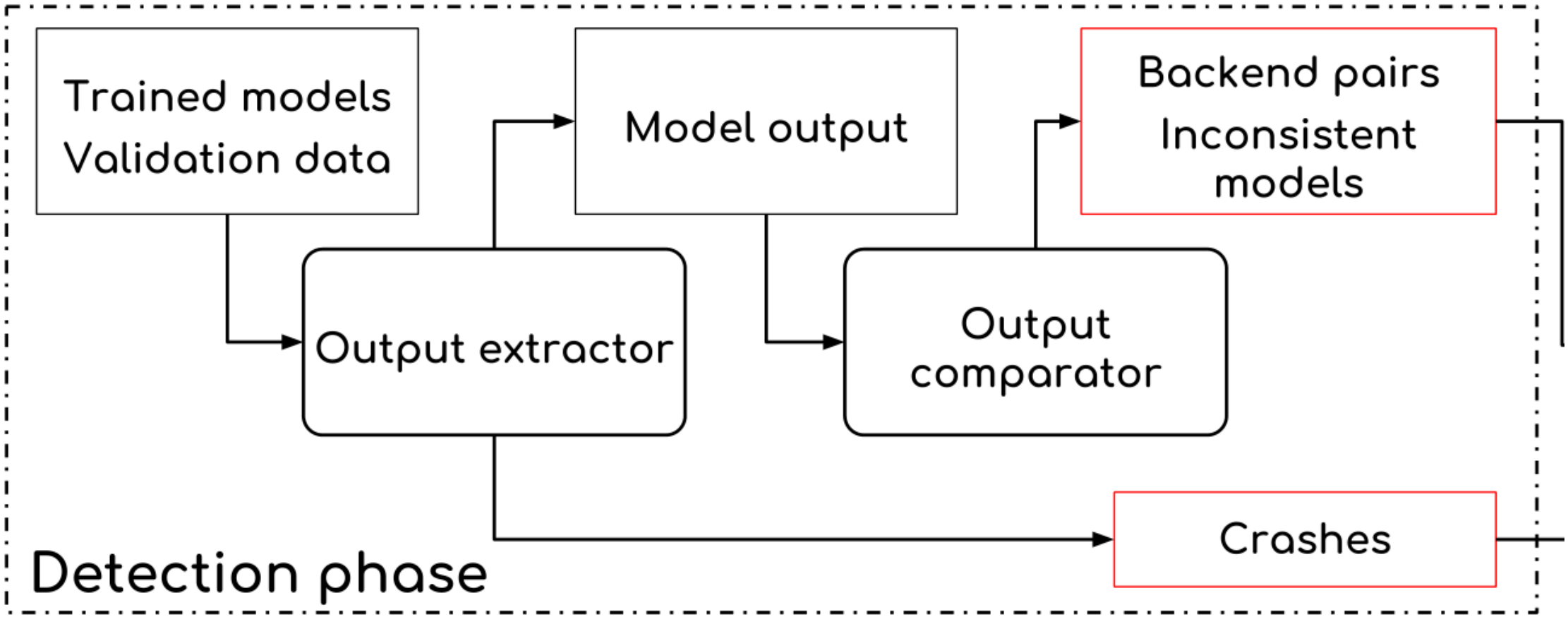
算法原理

- 算法流程
  - 检测阶段
  - 局部化阶段



算法原理

- 检测阶段


Trained models
Validation data
Model output
Backend pairs
Inconsistent models
Output extractor
Output comparator
Crashes
Detection phase

算法原理

- 输出不一致：
  - 如何区分暴露缺陷的不一致和无意义的不一致（bug-revealing inconsistencies、uninteresting inconsistencies）
    - 模型的结构、实际应用场景等不同，置信度变化区间不同

Top-1 confidence

LeNet1:0.85~0.90

Betago:0.45~0.60

  - 比较输出与真实标签的距离的差异
    - 计算在一组验证集上的Top-K准确率 → ✗
      - Top-K：预测结果中最有可能的K个结果是否包含有真实标签

$$logits = [0.1, 0.05, 0.1, 0.2, 0.35, 0.01, 0.03, 0.05, 0.01, 0.1]$$

Top-1:29.9%

Top-5:64.4%

TensorFlow　　CNTK

算法原理

- 距离度量
  - 基于分类的距离Class-based Distance (D_CLASS)
    - 根据真实标签在输出矩阵中排名的相对距离计算两个分类之间的距离
    - 给定分类模型的输出向量Y和真实标签C，得到分类的评分$\sigma_{C,Y}$

$$\sigma_{C,Y} = \begin{cases} 2^{k-rank_{C,Y}} & if\ rank_{C,Y} \le k \\ 0 & otherwise \end{cases}$$

$$C = \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_i \end{bmatrix} \rightarrow Y = \begin{bmatrix} p_1 \\ p_2 \\ \dots \\ p_i \end{bmatrix}$$

    - $rank_{C,Y}$为真实标签在分类结果中的排名, k设为5
    - 另一个后端框架输出向量为Y'，计算$\sigma_{C,Y'}$，得到$D_Class_{C,Y,Y'}$

$$D\_Class_{C,Y,Y'} = |\ \sigma_{C,Y} - \sigma_{C,Y'}\ |$$

    - 唯一不一致

1	2	3	4	5	6
16	15 - 8	7 - 4	3 - 2	1	0

算法原理

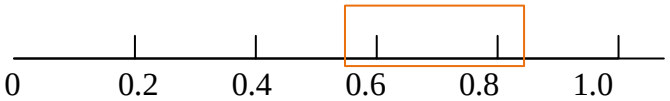
- 距离度量
  - 基于平均绝对误差的距离 Mean Absolute Deviation-based (D_MAD)
    - D_CLASS只用于分类模型，无法用于回归模型，D_MAD均适用
    - 给定真实标签O，模型输出Y，计算平均绝对距离$\delta_{Y,O}$

$$\delta_{Y,O} = \frac{1}{N}\sum_{i=1}^{N}|Y_i - O_i|$$

    - 另一个后端框架输出计算$\delta_{Y',O}$，计算D_MAD

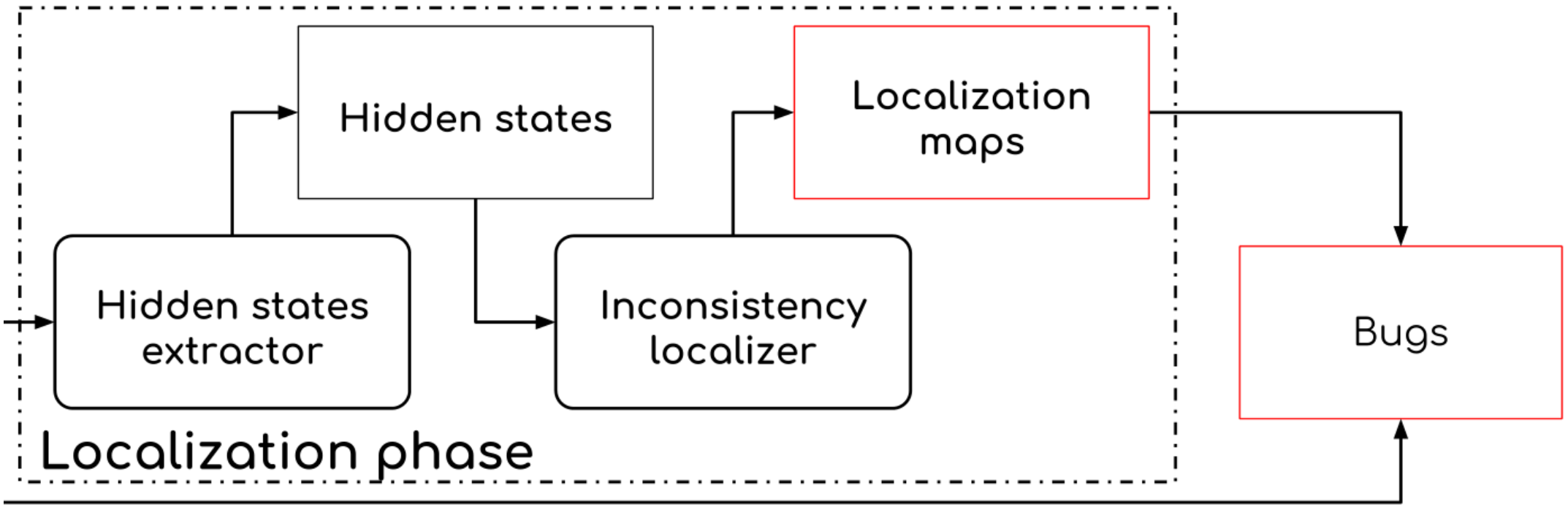
$$D\_MAD_{O,Y,Y',} = \frac{|\delta_{Y,O} - \delta_{Y',O}|}{\delta_{Y,O} + \delta_{Y',O}} \in (0,1)$$

$1 \rightarrow 100\%$ confidence　　$[100\%$ ... $0]$



算法原理

- 局部化阶段
  - 隐藏状态提取：获取模型网络每一层中间函数的输出
  - 不一致局部化：计算每对框架下每一层隐藏状态值的MAD距离



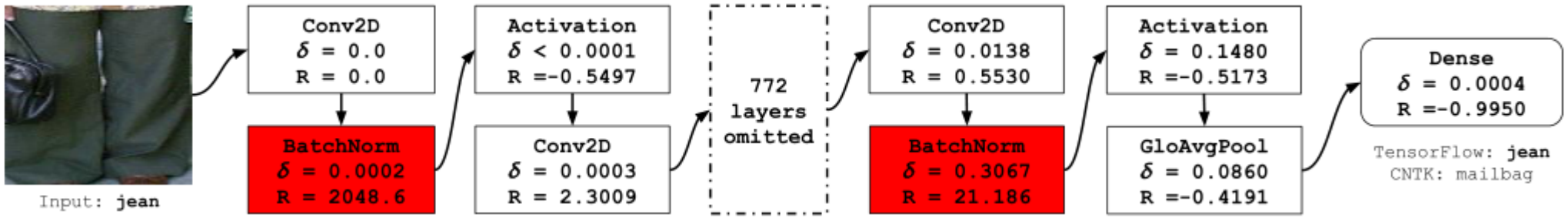
算法原理

- 不一致局部化
  - 两个框架下的模型L层的输出为S_L、S'_L，得到δ_{S_L,S'_L}
  - 误差会逐层传播，需要计算差异变化率
  - 对于输入$l \in pre(L)$，计算δ_{S_l,S'_l}，选取δ_{pre}

$$\delta_{pre} = \max(\delta_{S_l,S'_l})$$

  - 计算传播到层L的差异变化率R_L，手动分析过高的变化率

$$R_L = \frac{\delta_{S_L,S'_L} - \delta_{pre}}{\delta_{pre} + \varepsilon}$$



实验分析

- 输入
  - ImageNet
  - 自动驾驶
  - 手写数字识别
  - ……
- 待测软件库
  - 15个版本的Keras
  - CNTK、TensorFlow、Theano

任务	模型
ImageNet	Xception
	VGG16-19
	ResNet50
	InceptionV3
	InceptionResNetV2
	MobileNetV1-V2
	DenseNet121-169-201
	NASNetLarge-Mobile [8]
Self-driving models	DaveOrig-Norminit-Dropout
MNIST	LeNet1-4-5
Go game player	Betago
Anime faces recognition	AnimeFaces

实验分析

- 整体效果
  - 不一致数量104（361），7个bug

Root inconsistency	Localized layers (functions)	Affected backends	# Affected models	# Inc. bugs
Batch normalization	BatchNomalization	CNTK	11	2
Padding scheme	Conv2D, DepthwiseConv2D, SeparatableConv2D	TensorFlow, Theano	15	2
Pooling scheme	AveragePooling2D	Theano	3	1
Parameter organization	Trainable convolution	CNTK, Theano	18	2

  - 版本原因以外的崩溃：86/1173
    - 3个来自Keras，2个来自后端
  - 共计12个bug
    - 卷积核填充方案（Padding Scheme）
    - 池化方案（Pooling Scheme）
    - …

Dataset	Instances	# of Inconsistencies		
		TH-TF	TF-CN	CN-TH
ImageNet	5,000	10(34)	21(54)	18(46)
Driving	5,614		3(9)	3(12)
MNIST	10,000		3(9)	3(12)
Thai MNIST	1,665		1(3)	1(4)
KGS Go game	12,288	2(14)	3(12)	3(15)
Anime Faces	14,490	1(5)		1(6)
Dogs VS Cats	832		2(6)	2(8)
Dog species	835		3(8)	3(9)
Faces	466	2(14)	3(8)	6(15)
Pokedex	1,300	1(14)	1(3)	2(15)
GTSRB sign	12,630	2(14)	2(5)	2(7)
Total		18(95)	42(117)	44(149)
			104(361)	

实验分析

- 示例

- 两种距离的阈值设定T_C、T_{M}
  - $T_C = 8$、$T_{\mathrm{M}} = 0.2$
- 方法有效性验证
  - D_CLASS度量和Top-K精度

度量方法	D_CLASS（$k = 1$）	Top-1（$T_{AC} = 0\%$）
不一致发现数量	341	306

  - D_CLASS和D_MAD在分类模型上的效果

不一致判别	D_CLASS	D_MAD
唯一不一致发现数量	98	10

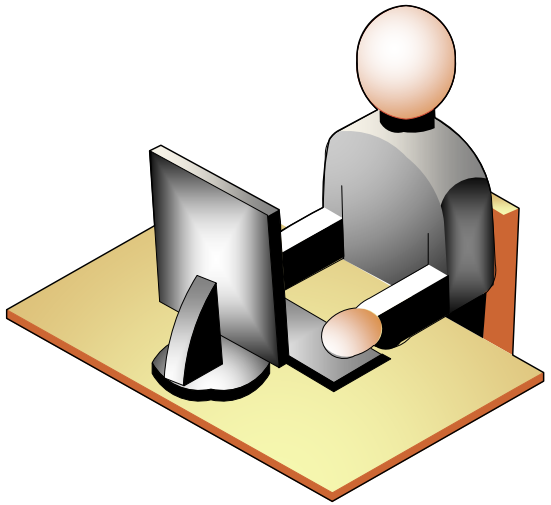
实验分析

- 优势：
  - 差分测试思想的引入很好地解决了测试预期的问题
  - 两种距离度量
- 局限性：
  - 输入局限于图像分类任务，覆盖库代码有限
    - 模型数量较少（20），构建大量模型费时费力
    - 公开模型往往经过无数次调整修复
  - 框架缺陷引起的不一致可能很小

优劣分析

优劣分析

- 优势：
  - 差分测试思想的应用
  - 不需要构建复杂的输入
- 劣势：
  - 覆盖库代码有限
  - 只观察了模型推理过程
  - 站在了“用户”的角度，而非测试人员



应用总结

- 未来发展
  - 制定更有效的模型生成策略，增加库代码覆盖率
  - 直接测试API函数
    - 如何自动/半自动构建复杂的输入

Args
input
filter
strides
padding
use_cudnn_on_gpu
explicit_paddings
data_format
dilations
name

参考文献

- **[1] H. V P, T. L, W. Q, et al. CRADLE: Cross-Backend Validation to Detect and Localize Bugs in Deep Learning Libraries[J].,2019:1027-1038.**

谢谢！

大成若缺，其用不弊。大盈若冲，其用不穷。大直若屈。大巧若拙。大辩若讷。静胜躁，寒胜热。清静为天下正。