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

神经网络模型测试方法与模型健壮性

硕士研究生 侯钰斌

2022年07月24日

内容提要

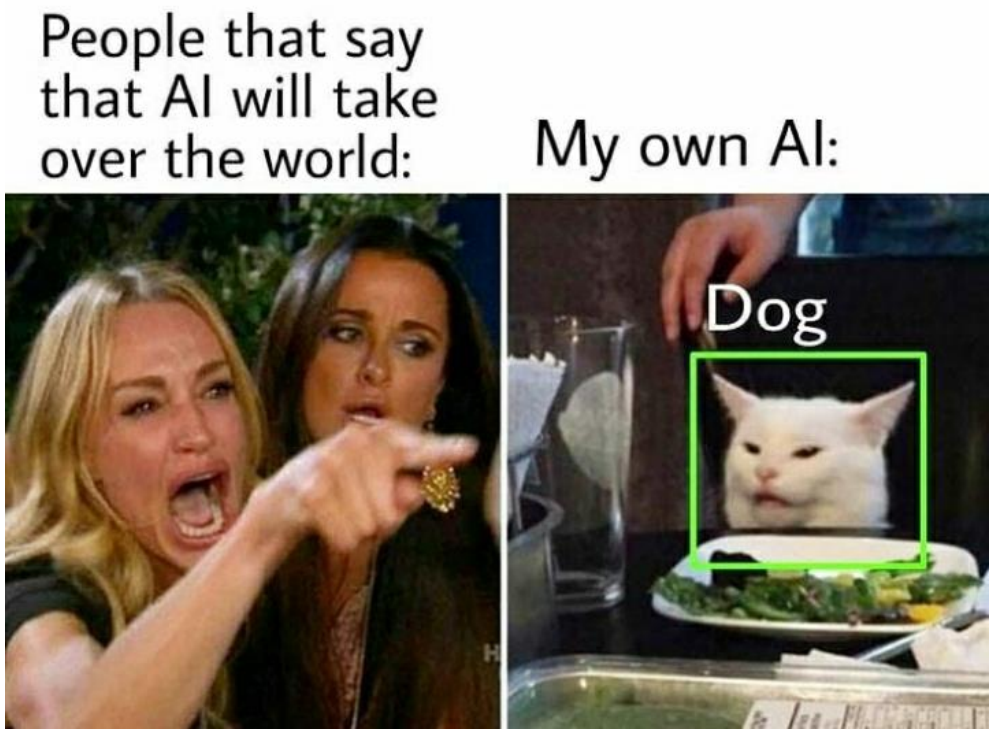
- 背景简介
- 基本概念
- 算法原理
- 实验结果与分析
- 应用总结
- 参考文献

预期收获

- 预期收获
  - 1. 了解神经网络测试的方法和框架
  - 2. 了解神经元覆盖率指导测试方法的不足
  - 3. 了解基于模型健壮性的神经网络测试的算法原理

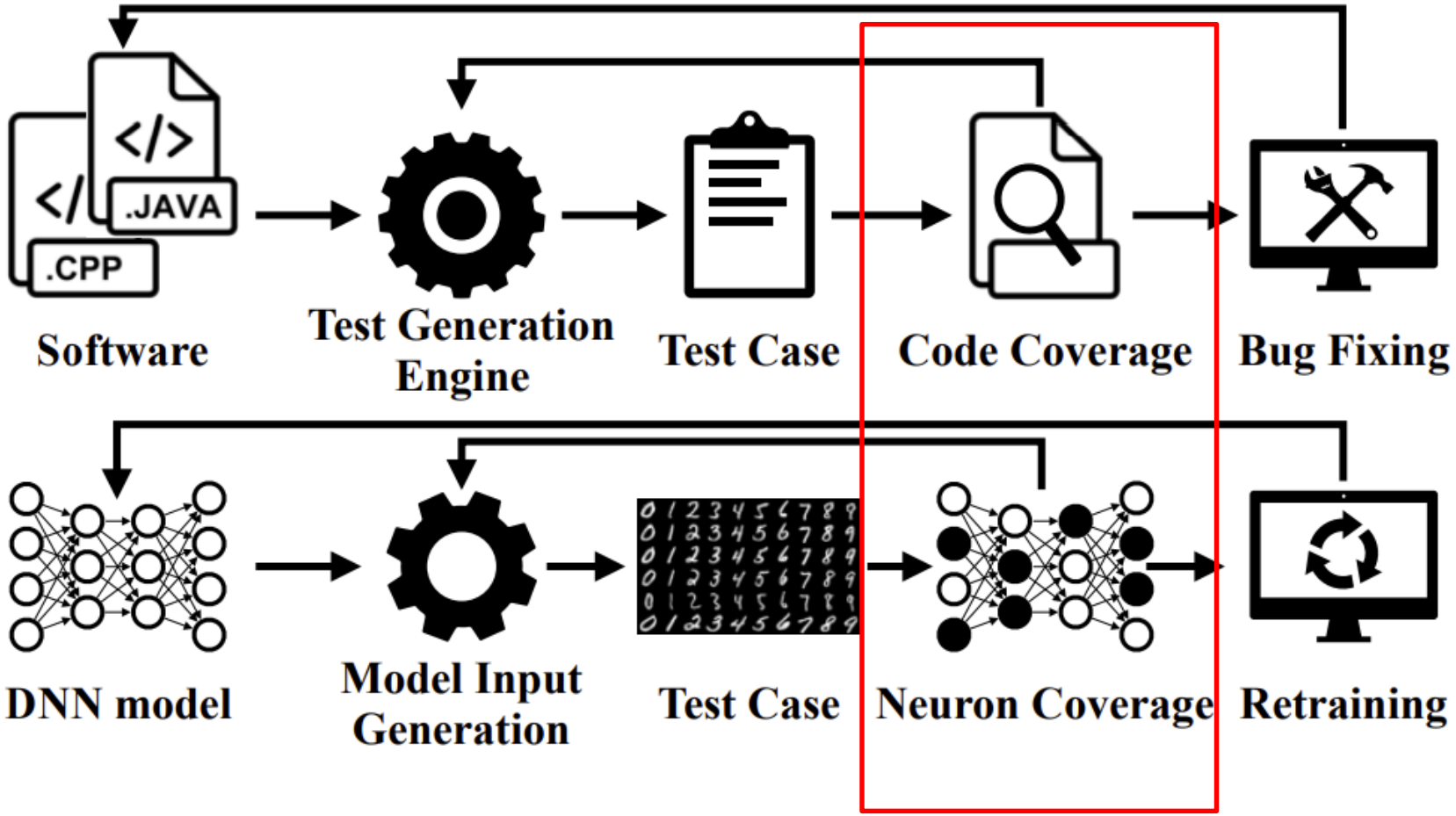
背景简介

- 人工智能的任务
  - 人工智能系统模拟人类学习的过程，学习数据到决策的抽象映射关系
  - 神经网络是最热门的人工智能模型之一
- 存在的问题
  - 开发者提供的数据是稀疏的，人工智能无法学习全部数据的特征
  - 人工智能学习的映射规则是难以解释的，也就无法对缺陷做出合理的解释
  - 环境与用户的干扰
  - 恶意攻击导致模型学习了错误的映射规则



背景简介

- 将软件测试流程迁移到神经网络模型测试任务
  - 为了体现测试充分性，需要建立一个与代码覆盖率类似的测试指标



基本概念

基本概念

- 健壮性
  - 数据或运行环境的扰动对模型正确性表现是否产生影响
  - 健壮性r的数学定义

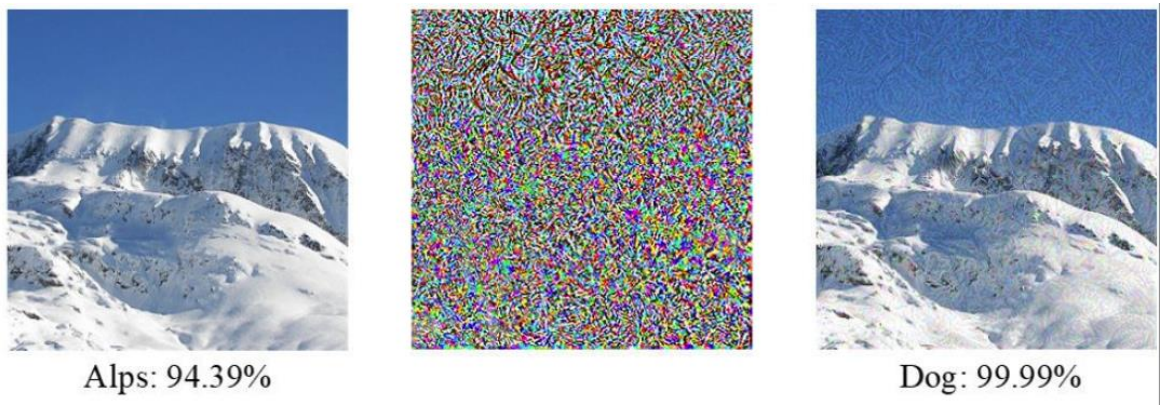
$$E(S) = Prediction_{x \sim D}[h(x) = c(x)]$$

$$r = E(S) - E(\delta(S))$$

  - D是数据集， x是该数据集中的数据， $h(x)$是模型预测的标签， $c(x)$是数据真实的标签
  - S是机器学习模型，$\delta(\cdot)$是模型环境或数据添加的扰动。$E(\cdot)$是模型的正确性

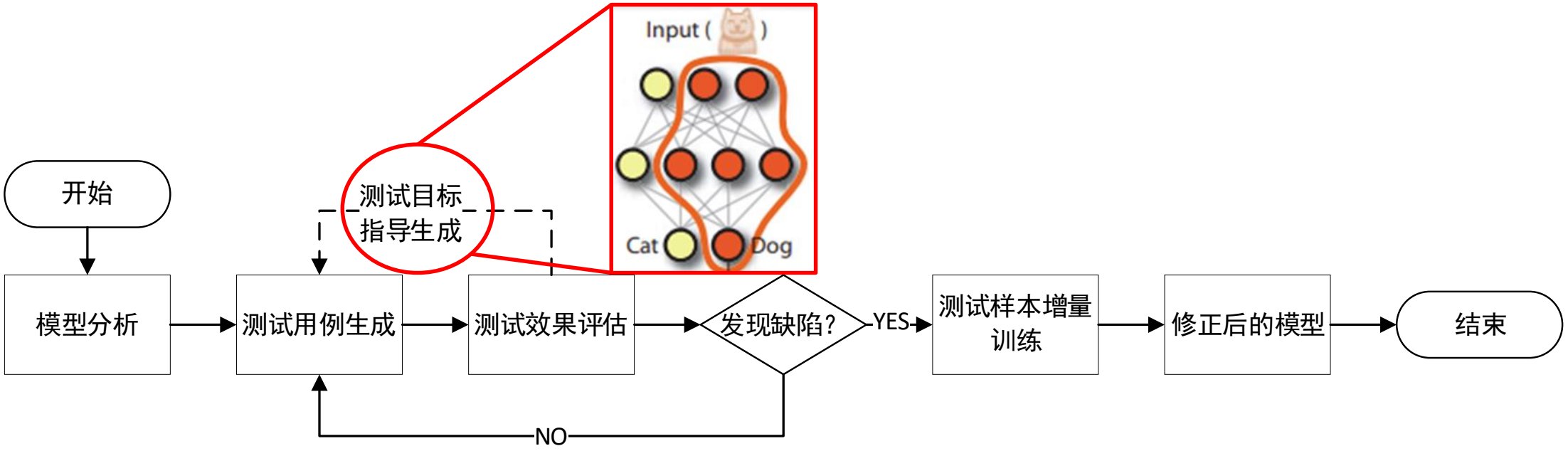
基本概念

- 模型健壮性的威胁
  - 对抗样本
    - 对输入样本故意添加一些人类无法察觉的细微的干扰，导致模型错误的决策输出
  - 后门攻击
    - 对数据分布进行修改，使模型对某类输入特征十分敏感，在特定输入下触发后门导致模型产生错误决策
    - 对存在于内存中模型二进制值进行修改，使模型对某类特征更加敏感



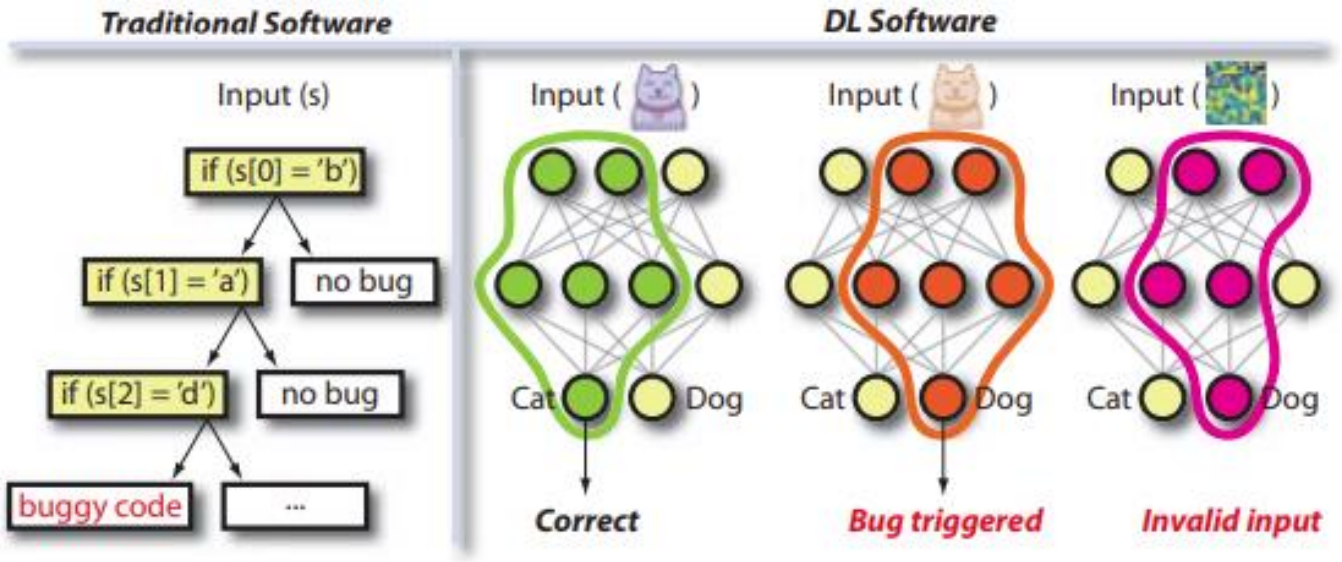
基本概念

- 发现影响模型健壮性的缺陷
  - 依照模糊测试的流程，为测试样本添加扰动生成新的测试样本
  - 如果生成的测试样本导致模型决策错误，则保留当前样本，否则继续测试
  - 使用所有触发模型决策错误的样本进行增量训练，得到修正的模型



基本概念

- 代码覆盖率
  - 不同的输入会满足不同的判定条件，触发不同的程序路径，导致不同的输出
- 神经元覆盖率
  - 前一个神经元的输出会影响后一个神经元的状态
  - 不同的样本特征使不同神经元被激活，导致模型决策结果不同



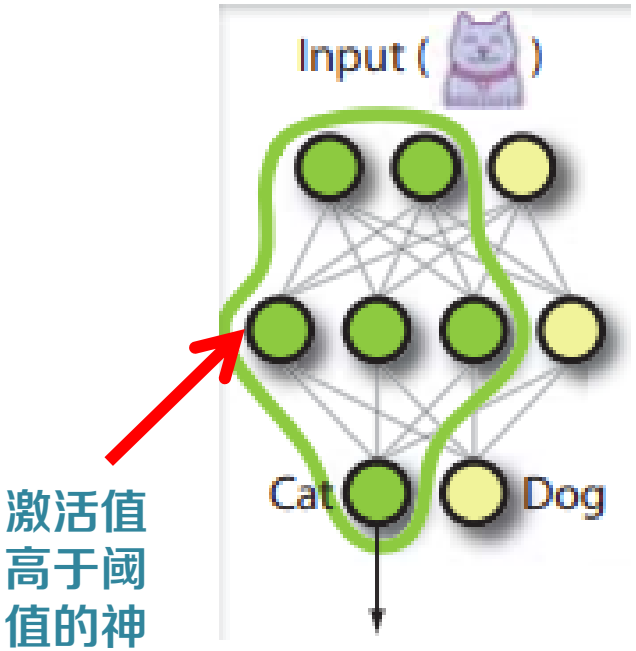
代码覆盖和神经元覆盖示意

基本概念

- 神经元覆盖率的定义
  - 一个测试样本所覆盖的神经元数量与神经网络神经元总数的比值
  - 若神经元的激活值高于设定阈值，则认定神经元被激活

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

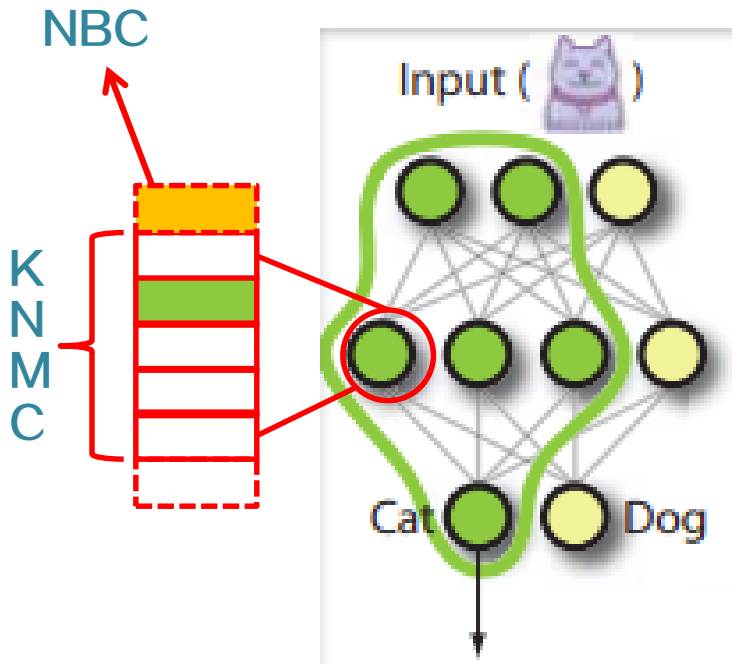
  - $T=\{x_1,x_2,\dots\}$，表示输入的测试样本集
  - $N=\{n_1,n_2,\dots\}$，表示神经网络中神经元的集合
  - $out(n,x)$，对于某个输入x，神经元n的激活值，t是设定阈值



神经元覆盖率

基本概念

- K阶神经元覆盖率（KMNC）
  - 获取模型中每个神经元在训练集上的激活值范围，并将值域分为k段（红色实线）
  - 当样本输入模型后，神经元激活值处于某个值域，则认为该值域被覆盖（绿色填充）
  - 神经元存在k种激活状态，能更加丰富地表示模型决策的状态
- 神经元边界覆盖率（NBC）
  - 当神经元激活值超出训练集的激活值范围时，该神经元被覆盖（红色虚线与橙色填充）
  - 在这种情况下，神经元的输出不正常，很可能触发模型缺陷



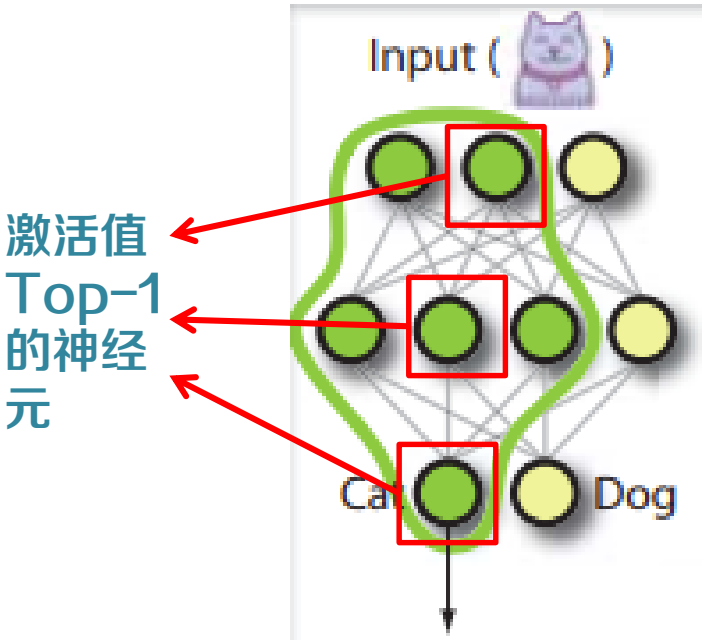
KNMC&NBC

基本概念

- Top-k神经元覆盖率
  - 模型N有l个隐藏层，在输入x的情况下，第i层$(1 \leq i \leq l)$中选择k个输出值最大的神经元，统计被测数据集中所有成为最活跃的k个神经元的数目和所有神经元数目的比值

$$TKNCov(T,k) = \frac{|\cup_{x \in T} (\cup_{1 \leq i \leq l} top_k(x,i))|}{|N|}$$

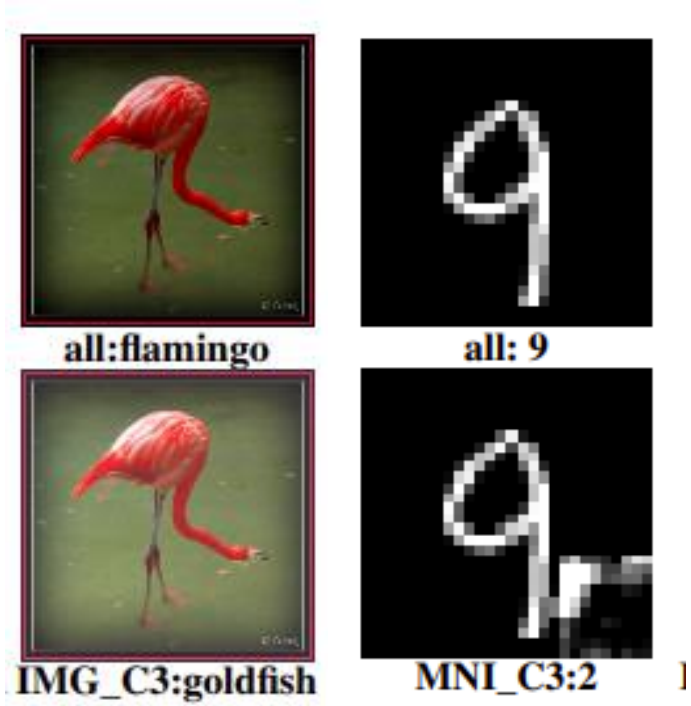
  - $top_k(x,i)$是在输入x下，模型N在第i个隐藏层中输出值最大的k个神经元，所有隐藏层的top_k可以认为是模型在输入x下的激活形式



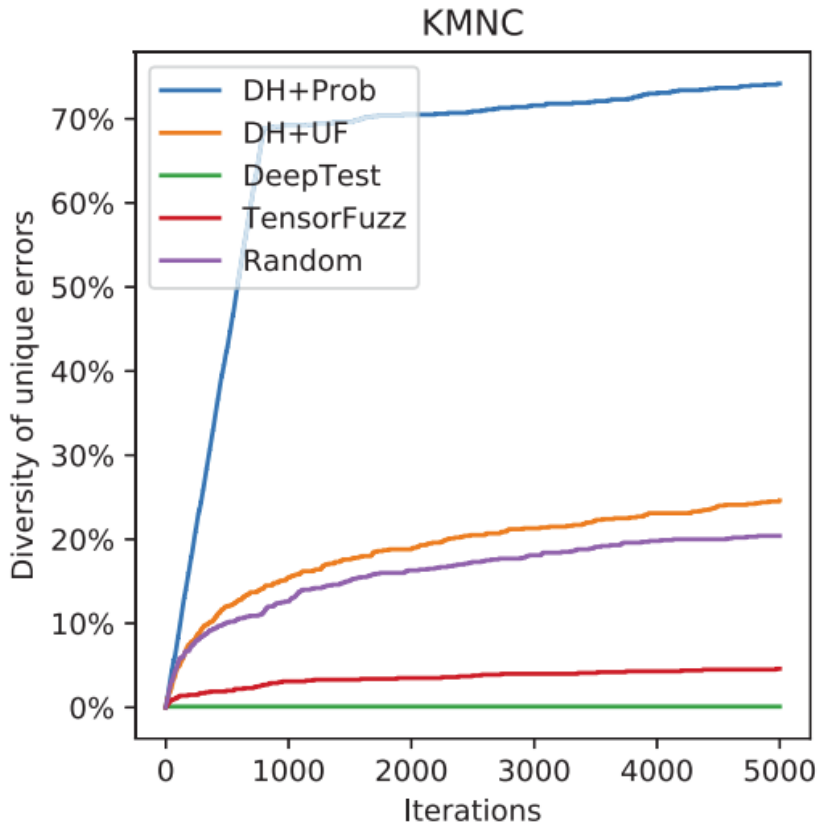
Top-k神经元覆盖率

基本概念

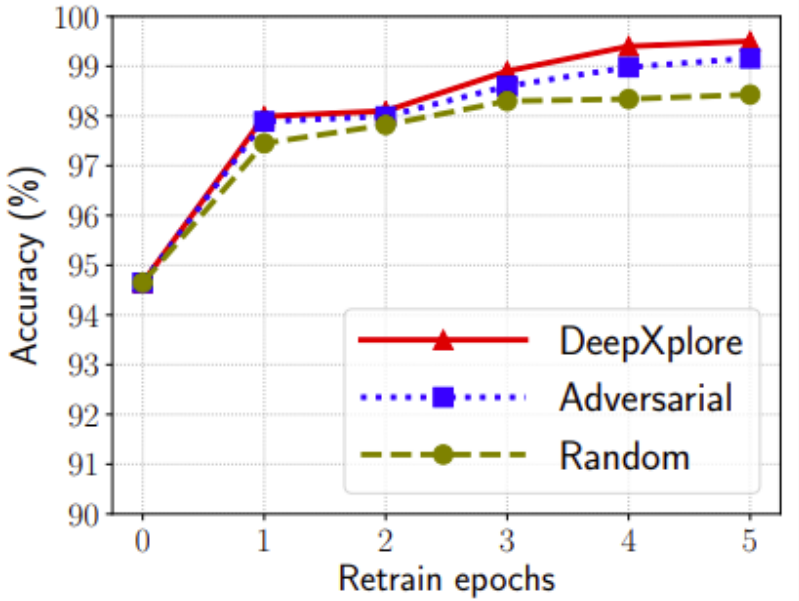
- 神经元覆盖率测试效果
  - 发现大量导致模型决策错误的样本
  - 测试样本可用于改进模型健壮性



导致模型决策错误的样本



测试中触发模型决策错误的频率



模型改进效果

算法原理

算法原理

T	研究当前 DNN 模型测试方法对模型质量提升的效果
I	DNN分类模型、图像数据集
P	1. 使用神经元覆盖率指导测试，生成测试样本组 2. 使用生成的测试样本增量训练被测模型 3. 检查模型的健壮性指标改进情况
O	模型质量提升定量效果

P	神经元覆盖率指导测试与模型质量提升的相关性如何
C	分类神经网络模型，白盒测试
D	如何通过数据分析测试的效果并解释理论因素
L	2020 ESEC/FSE（CCF A）

算法原理

- 对抗样本的模型准确率指标
  - 错误分类占比，N 表示所有样本数量，x_i^a 表示生成样本，y_i 表示原始样本结果，$F(x_i^a)$ 表示生成样本在模型中的结果，指标越小说明模型健壮性越好：

$$MR = \frac{1}{N}\sum_{i=1}^{N} count(F(x_i^a) \neq y_i)$$

  - 对抗样本平均置信度，n 表示结果错误的样本数量，$P(x_i^a)_{F(x_i^a)}$表示生成样本在输出结果错误时置信度，指标越小说明模型的健壮性越好：

$$ACAC = \frac{1}{n}\sum_{i=1}^{n} P(x_i^a)_{F(x_i^a)}$$

  - 正确样本平均置信度，指标越大说明模型的健壮性越好：

$$ACTC = \frac{1}{n}\sum_{i=1}^{n} P(x_i^a)_{y_i}$$

算法原理

- 对抗样本扰动指标
  - 平均 L_p扰动距离，x_i是原始样本，指标越小说明生成对抗样本的扰动越小：

$$ALD_p = \frac{1}{n}\sum_{i=1}^{n}\frac{|x_i^a - x_i|_p}{|x_i|_p}$$

  - 平均结构度，$SSIM$量化两个图片的结构度，指标越大说明生成的样本与原始样本越相似：

$$ASS = \frac{1}{n}\sum_{i=1}^{n} SSIM(x_i^a - x_i)$$

算法原理

- 对抗样本的健壮性指标
  - 错误置信度，$\underset{F(x_i^a)}{\max}(P(x_i^a))$ 表示在生成样本的输出结果中第二大置信度，指标越大说明样本的对抗性越大：

$$NTE = \frac{1}{n}\sum_{i=1}^{n}[P(x_i^a)_{F(x_i^a)} - \underset{F(x_i^a)}{\max}(P(x_i^a))]$$

  - 图像压缩健壮性，IC 表示图像压缩处理，指标越大说明样本的对抗性越大：

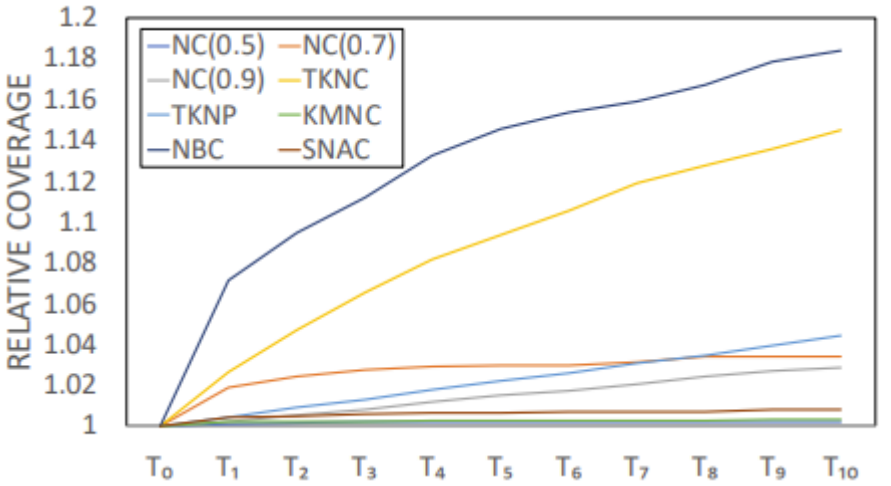
$$RIC = \frac{count(F(IC(x_i^a)) \neq y_i)}{count(F(x_i^a) \neq y_i)}$$

算法原理

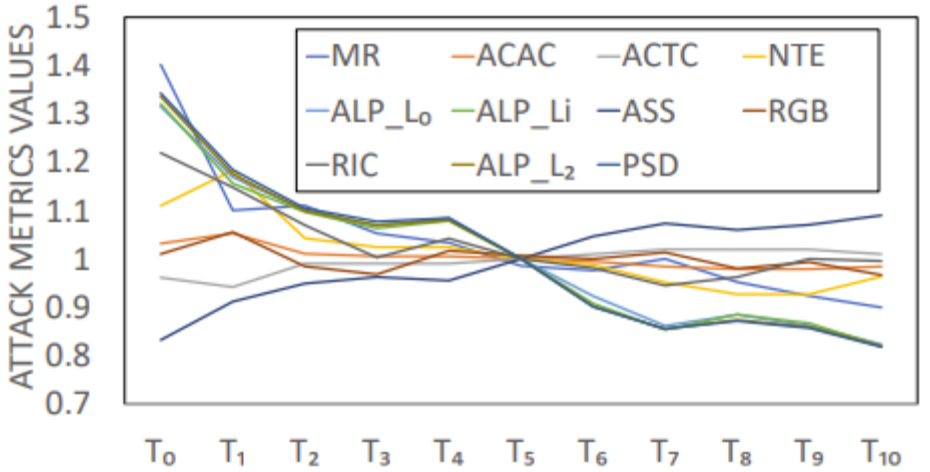
- 研究问题1：神经元覆盖率指标与模型健壮性的相关性
  - 覆盖率越大测试样本，对模型健壮性指标的影响越大?
- 实验设计
  - 使用对抗样本生成方法生成基线重训练样本集 T_0
  - 使用覆盖率测试方法生成使覆盖率指标显著提升的样本集
  - 向 T_0 中添加覆盖率测试生成的样本集，得到样本T_1
  - 重复第二步并将最新生成的样本添加到上一轮构造的样本集 T_i 中，得到更多的重训练样本集 $T_0, T_1, T_2, T_3, \ldots\ldots$
  - 将样本集输入被测模型得到实验指标数据

实验结果

- 研究问题1：神经元覆盖率指标与模型健壮性的相关性
  - 加入越多覆盖率大的样本，数据集的覆盖率越大，但上升速率变慢，原因是选择的样本覆盖率提升越来越小
  - 加入越多覆盖率大的样本，数据集在模型准确率指标、样本扰动指标、样本健壮性指标上没有明显的变化趋势
  - 覆盖率指导生成的样本与模型健壮性没有明显的相关性



覆盖率与数据集的关系



覆盖率与实验指标的关系

算法原理

- 研究问题2：覆盖率测试方法对探索模型缺陷的效果
  - 覆盖率测试方法能否有效地探索导致模型决策错误的缺陷?
- 实验设计
  - 使用DeepHunter和PGD攻击分别生成测试样本 D_H 和 D_P
    - DeepHunter使用对提升覆盖率有帮助的样本修改方法生成新的样本
  - 比较D_H 和 D_P 中导致模型决策错误的样本数量
  - 比较D_H 和 D_P 中生成样本与原始样本的相似程度
  - 检查D_H 中生成的导致模型决策错误的样本数量与覆盖率的关系

实验结果

- 研究问题2：覆盖率测试方法对探索模型缺陷的效果
  - 使用DeepHunter和PGD生成样本对模型进行测试。PGD的生成样本触发模型缺陷的效果高于DeepHunter，且添加的扰动小。
  - DeepHunter的扰动大的原因是其生成样本使用了仿射变换算子，对样本原始语义改变较大

Dataset	Model	MR		ℓ_∞	
		D_H	D_P	D_H	D_P
MNIST	LeNet-1	92.5%	100%	0.986	0.3
	LeNet-4	90.0%	100%	0.983	0.3
	LeNet-5	84.3%	100%	0.963	0.3
CIFAR	VGG-16	94.4%	88.7%	3.114	0.031
	ResNet-20	94.4%	99.8%	3.242	0.031
SVHN	SADL-1	61.7%	100%	0.689	0.031
	SADL-2	92.4%	99.9%	0.714	0.031
	SADL-3	64.4%	100%	0.675	0.031

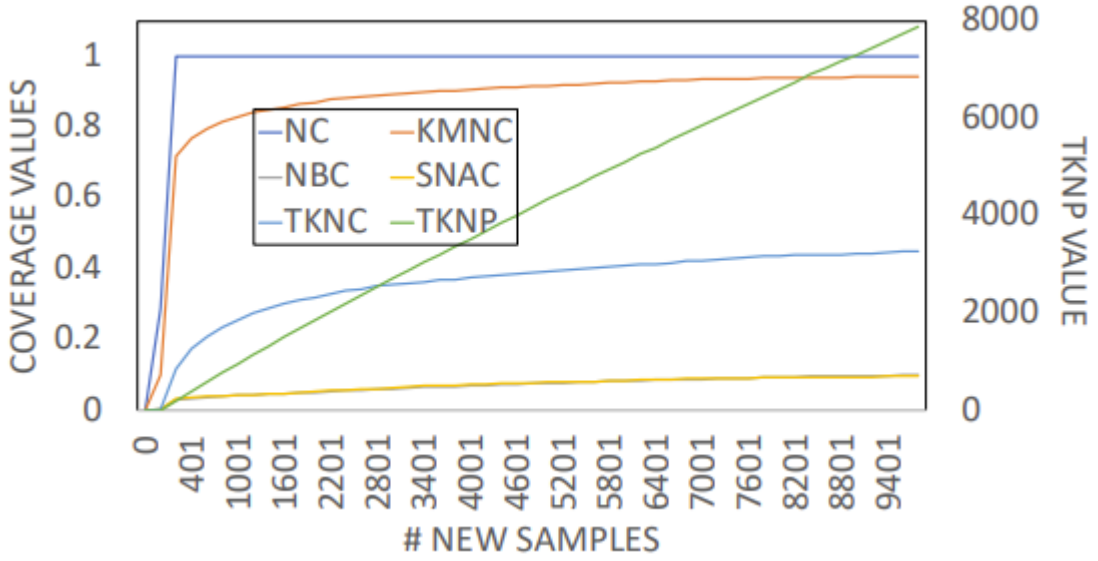
两种方法的实验效果

(a) Original Images (b) DeepHunter (c) PGD

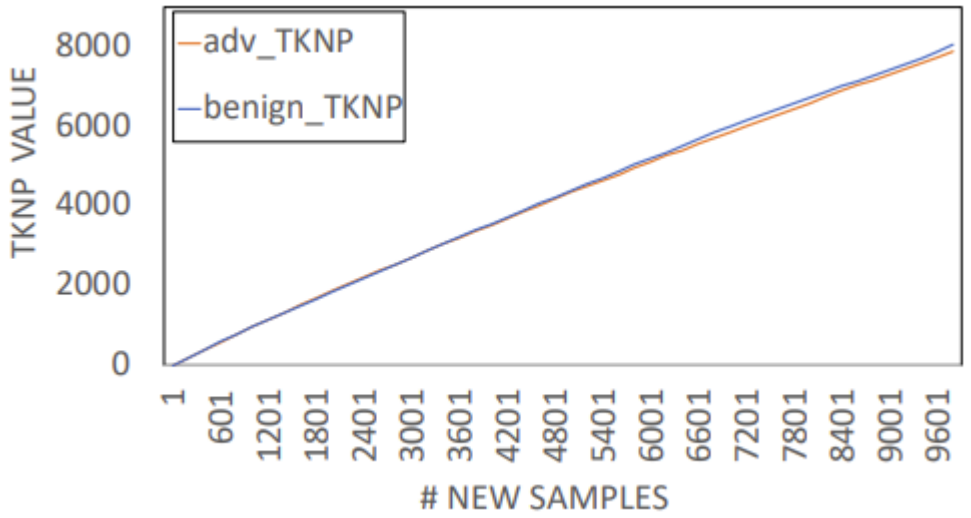
两种方法生成样本对比

实验结果

- 研究问题2：覆盖率测试方法对探索模型缺陷的效果
  - PGD生成的对抗样本能增大各类覆盖率指标，但是生成样本初期覆盖率提升快，后期生成样本的覆盖率基本不变，说明覆盖率增大并不能完全表示模型多样化的激活状态，也不能完全探索模型的缺陷
  - Top-k神经元覆盖率指导样本生成实验说明该指标对模型缺陷不敏感



PGD对抗样本与覆盖率的关系图



Top-k神经元覆盖率
生成正常与对抗样本数目

算法原理

- 研究问题3：覆盖率测试生成的样本对模型健壮性提升的效果
  - 使用覆盖率测试的生成的样本与对抗攻击生成的样本对模型进行重训练，哪一种方法使模型的健壮性提升效果更好?
- 实验设计
  - 使用覆盖率测试方法生成样本并添加到原始训练样本中
  - 使用混合数据集重训练模型
  - 使用PGD攻击方法生成样本混入原始训练集重训练模型作为基线
  - 检查重训练模型对对抗样本的检测准确率

实验结果

- 研究问题3：覆盖率测试生成的样本对模型健壮性提升的效果
  - 对抗训练比较难以收敛，所以会产生重训练后模型性能变低的情况
  - 简单地对抗训练不一定会提升模型的性能和健壮性
  - 使用覆盖率测试生成样本重训练模型后，对PGD对抗样本的识别准确率远低于使用覆盖率测试生成样本，所以覆盖率测试生成样本对模型健壮性提升效果有限

Dataset	Model	Benign	D_H	PGD
MNIST	LeNet-1	98.7%(+0.09%)	90.3%(+2.37%)	0%(+0%)
	LeNet-4	98.7%(+0.07%)	90.1%(+2.3%)	0%(+0%)
	LeNet-5	98.71%(-0.26%)	91%(+1.1%)	0%(+0%)
	LeNet-1 Adv.	97.6%(-1.07%)	91.3%(+3.4%)	19.2%(+19.2%)
	LeNet-4 Adv.	96.9%(-1.72%)	88.9%(+1.1%)	9.6%(+9.6%)
	LeNet-5 Adv.	97%(-1.97%)	90.1%(+0.2%)	30.5%(30.3%)
CIFAR	VGG-16	10%(-82.8%)	9.89%(-47.41%)	9.99%(+8.8%)
	ResNet-20	75.4%(-16.4%)	60.5%(+1.4%)	0.13%(+0.13%)
	VGG-16 Adv.	87.8%(-5%)	55.9%(-1.4%)	40.9%(+39.7%)
	ResNet-20 Adv.	86.7%(-5.04%)	57.5%(-1.7%)	36.6%(+36.6%)
SVHN	SADL-1	93.97%(+4.27%)	82.95%(+6.9%)	0%(+0%)
	SADL-2	91.55%(+3.83%)	77.65%(+5.9%)	0.1%(+0.1%)
	SADL-3	94.28%(+1.71%)	84.47%(+5.1%)	1.1%(+1.1%)
	SADL-1 Adv.	85.8%(-3.9%)	71.25%(-4.8%)	46.6%(+46.6%)
	SADL-2 Adv.	81.7%(-6.12%)	66.73%(-4.8%)	44.6%(+44.6%)
	SADL-3 Adv.	87.7%(-4.87%)	74.13%(-5.23%)	50.6%(+50.6%)

重训练模型与原始模型样本识别准确率
Adv指使用PGD生成样本重训练
无Adv指使用覆盖率测试生成样本重训练

实验结论

- 结论
  - 神经元覆盖率与传统代码覆盖率不一致，即使测试指标达到高覆盖率也存在大量使模型决策错误的样本，所以文中提及的覆盖率指导测试的方法的有效性存疑
  - 测试需要利用模型多模态的结构信息和多种测试指标
- 个人对文中覆盖率表现的解释
  - 覆盖率存在明显的上界，无法全面完整地表示模型在样本空间下的决策状态
  - 覆盖率只考虑了神经元激活值的表现形式，没有考虑被激活神经元在空间上组合形式

算法原理

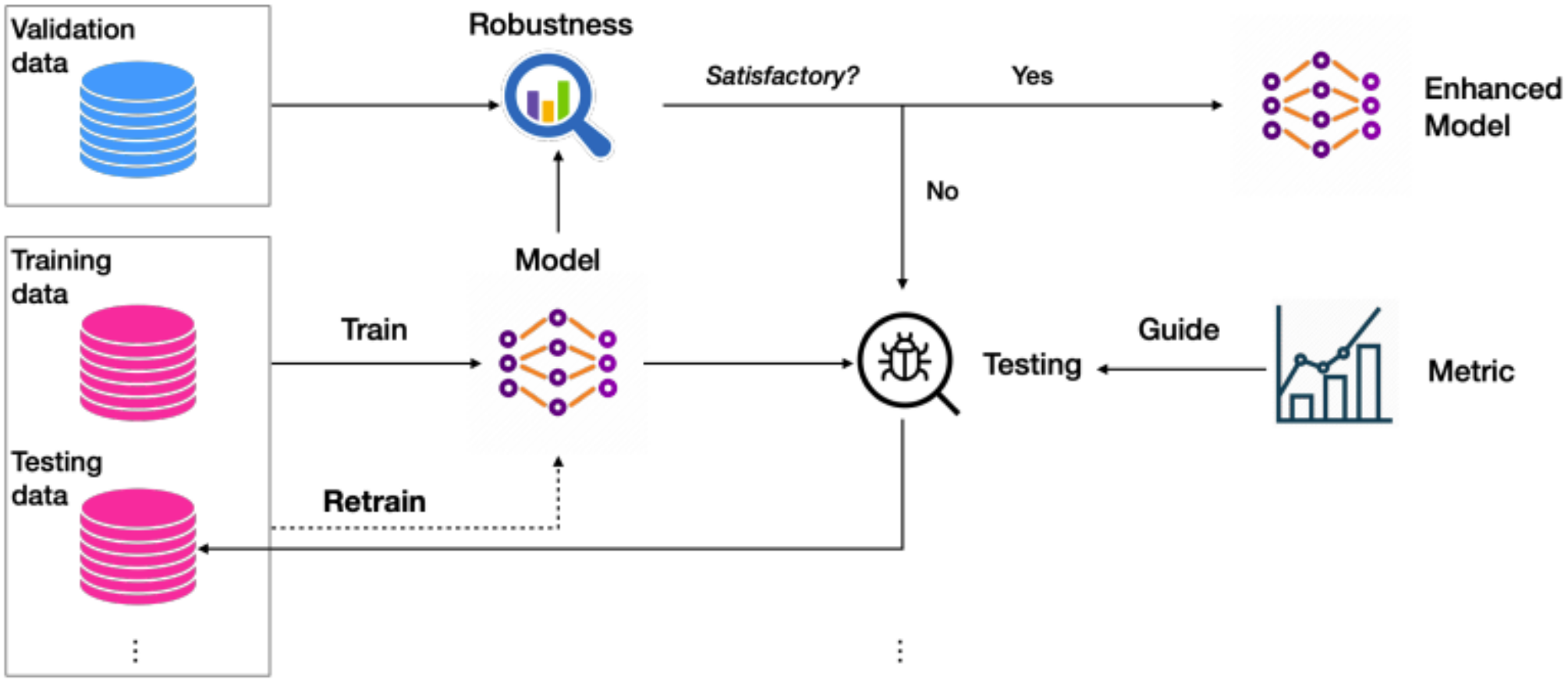
算法原理

T	从模型健壮性的角度进行自动化测试
I	被测模型、训练样本和测试样本
P	1. 计算当前样本的输出置信度和健壮性指标作为测试目标 2. 计算测试目标对样本的梯度，使用梯度上升生成样本 3. 如果样本的健壮性指标高于阈值则保存，否则继续迭代 4. 遍历提供的原始样本集，得到用于重训练的样本
O	触发被测模型分类错误且对模型健壮性影响大的样本

P	神经元覆盖率指导测试与模型健壮性的相关性小
C	分类神经网络模型的白盒测试
D	如何使用较为简单的指标表示样本对模型的健壮性的影响
L	2021 ICSE（CCF A）

算法原理

- Robot流程图



算法原理

- 对抗训练的目标是降低模型对对抗样本的决策错误率
  - 对抗样本的目标是在一定扰动范围内增大模型对输入样本的决策结果与正确结果的差距，即提升该样本在模型上的训练损失
  - 对抗训练的目标是减小所有对抗样本的上述损失
  - 在限制条件下，生成样本的损失越大，样本的针对模型的对抗性越大

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \max_{||x_i' - x_i||_p \leq \epsilon} L(f(\theta, x_i'), y_i)$$

- θ 是模型参数，f 是深度学习模型，x_i 是输入样本，y_i是模型决策输出，L是模型训练的损失函数，x_i' 是添加扰动后的样本，与原始样本的距离不超过 ϵ

算法原理

- 样本对模型的对抗性
  - 如果根据样本的梯度方向修改样本，样本在当前模型下的损失会增大并最终收敛，这样修改样本能最快提升其对抗性
  - 为了度量样本的对抗性，认为样本损失的收敛速度越快，则样本的潜在对抗性越强，则有以下定义，对于样本 x_0，其邻域为$X = \{x| \ ||x - x_0||_p \leq \epsilon\}$，基于 x_0 生成的样本 x_t，计算FOSC（一阶平稳条件）值：

$$FOSC(x_t) = \max_{x \in X} < x - x_t, \nabla_x L(f(\theta, x_t), y_i) >$$
$$= \epsilon ||\nabla_x L(f(\theta, x_t)), y_i)||_2$$

  - FOSC值越小，则样本的损失能最快收敛到最大值，样本对抗性越大
  - 模型的健壮性越高，使模型决策错误的对抗样本的对抗性越大，FOSC越小
  - 该指标筛选出与模型健壮性有关的样本，减小对抗训练的性能花费

算法原理

- 测试过程解析
  - 测试目标为对抗性低且被分类错误的样本

$$obj = \sum_{i=2}^{k} P(c_i) - P(c_1) + \alpha * FOSC(x)$$

  - $P(c_1)$是模型输出的最高置信度，$P(c_i)$ 为其他置信度
  - 使用梯度上升生成新的测试样本
  - 对抗性较大和较小的样本都可以继续作为测试种子样本继续迭代生成更多的样本
  - 保存当前被分类错误的样本

Algorithm 3 FOL-Fuzz($f, seeds\_list, \epsilon, \xi, k, \lambda, iters$)

```
Let fuzz_result = ∅
for seed ∈ seeds_list do
    Maintain a list s_list = [seed]
    while s_list is not empty do
        Obtain a seed x = s_list.pop()
        Obtain the label of the seed c1 = f(x)
        Let x' = x
        for iter = 0 to iters do
            Set optimization objective obj using Eq. 8
            Obtain grads = ∇obj / ∇x'
            Obtain perb = processing(grads)
            Let x' = x' + perb
            Let c' = f(x')
            Let dis = Dist(x', x)
            if FOL(x') ≥ FOL_m and dis ≤ ε then
                FOL_m = FOL(x')
                s_list.append(x')
                if c'! = c1 then
                    fuzz_result.append(x')
                end if
            end if
            if FOL(x') < ξ and dis ≤ ε then
                s_list.append(x')
                if c'! = c1 then
                    fuzz_result.append(x')
                end if
            end if
        end for
    end while
end for
```

实验结果

- 实验准备

Dataset	Training	Testing	Model	Accuracy
MNIST	60000	10000	LeNet-5	99.02%
Fashion-MNIST	60000	10000	LeNet-5	90.70%
SVHN	73257	26032	LeNet-5	88.84%
CIFAR10	50000	10000	ResNet-20	90.39%

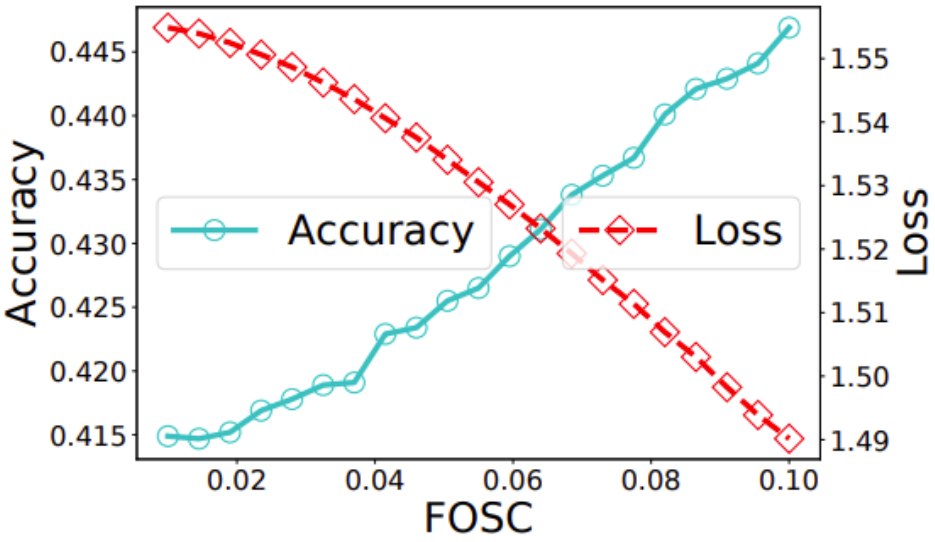
实验数据集

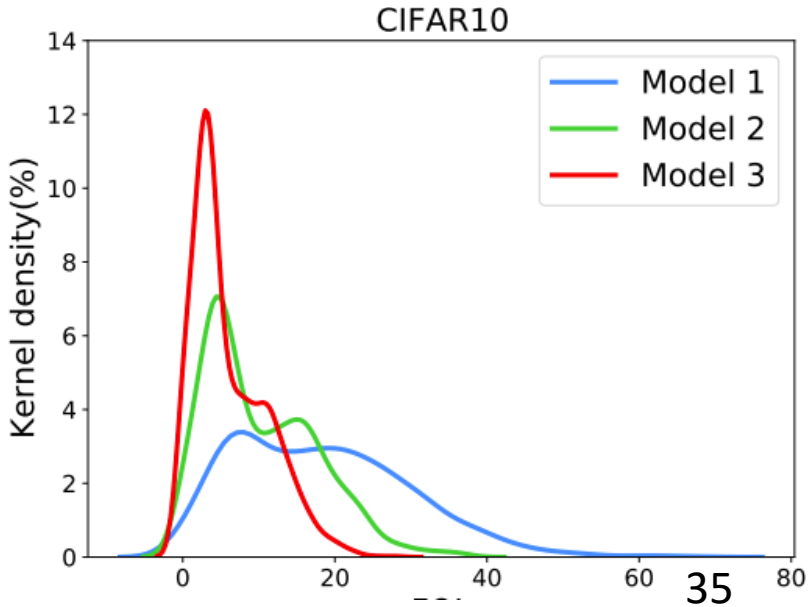
Testing Method	Parameter	MNIST	SVHN	Fashion-MNIST	CIFAR10
FGSM	Step size	0.3	0.03	0.03	0.01
PGD	Steps	10	10	10	10
	Step size	0.3/6	0.03/6	0.3/6	0.01/6
DeepXplore	Relu threshold	0.5	0.5	0.5	0.5
DLFuzz/ADAPT	Time per seed	10 s	10 s	10 s	20 s
	Relu threshold	0.5	0.5	0.5	0.5

对比方法与参数

实验结果

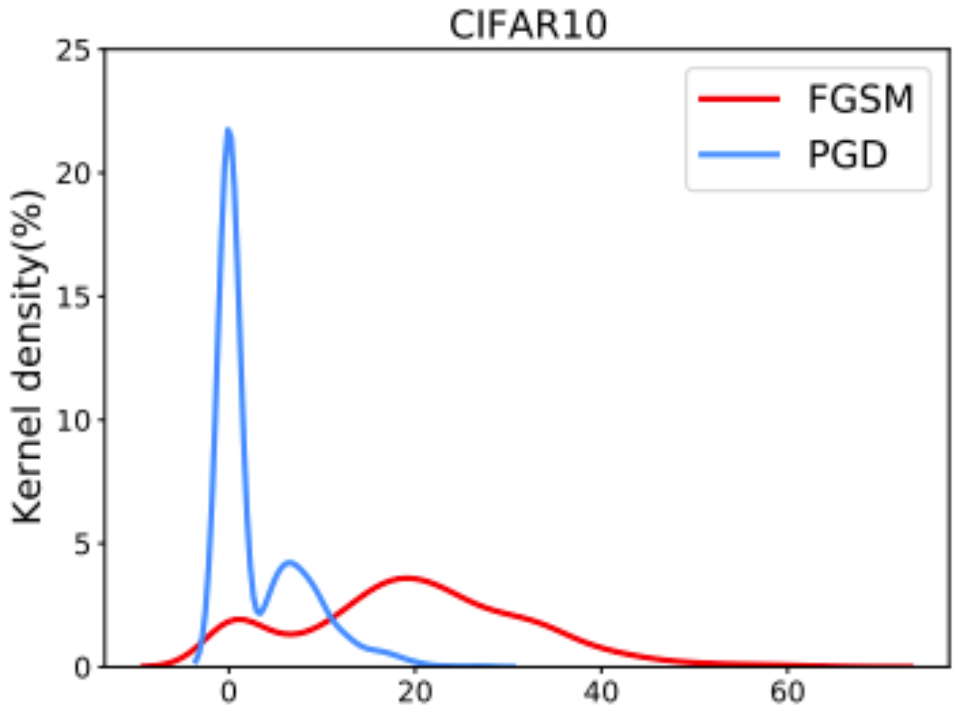
- FOSC与模型健壮性的相关性
  - 选择三个模型，model1是未经对抗训练的模型，model2是使用100个对抗样本训练后模型，model3是使用200个对抗样本训练后的模型，模型健壮性依次递增
  - 使用PGD攻击方法生成相同数量的对抗样本并算样本的FOSC值分布
- 越健壮的模型，使其决策错误的对抗样本的对抗性越高，FOSC值越小，说明FOSC与模型健壮性密切相关





实验结果

- FOSC与对抗攻击的有效性的关系
  - 针对相同模型，使用FGSM和PGD生成相同数量的对抗样本并计算样本的FOSC值分布
  - PGD的攻击效果比FGSM的攻击效果更好
  - PGD生成的对抗样本FOSC值更小
- 生成对抗样本方法的效果越好，其样本的FOSC值越小



实验结果

- 与其他覆盖率测试的效果对比
  - 相同时间内使用最先进的覆盖率测试和本文方法生成样本，并使用生成样本对模型进行对抗训练，计算重训练前后模型对重训练样本的FOSC值
  - 相同时间内该方法生成测试样本数量较少
  - 使用本文方法生成的样本对模型进行对抗训练，相比原始模型，训练后的模型健壮性提升幅度更大
- 该方法生成的测试样本对模型健壮性提升更大，效率更高

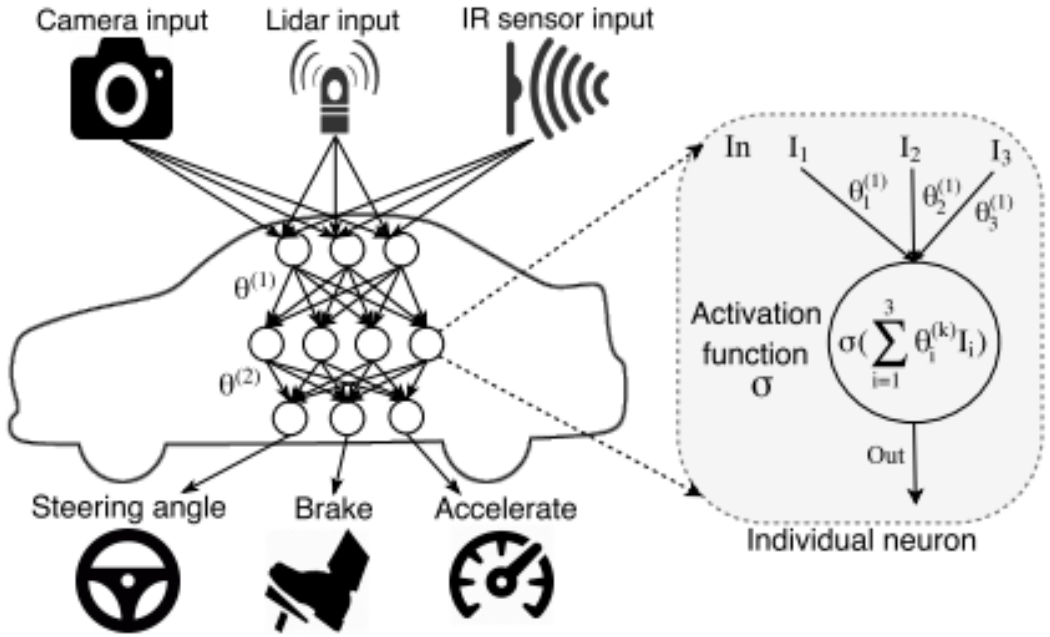
Dataset	5 min		10 min		20 min	
	# Test case	Robustness↑	# Test case	Robustness↑	# Test case	Robustness↑
MNIST	1692/2125	33.62%/18.73%	3472/4521	48.04%/36.46%	7226/8943	68.02%/54.38%
Fashion-MNIST	4294/5485	40.75%/6.74%	8906/10433	53.88%/14.94%	18527/21872	69.03%/27.24%
SVHN	6236/8401	24.25%/21.3%	12465/17429	30.42%/27.52%	24864/33692	39.99%/34.51%
CIFAR10	1029/1911	18.62%/17.03%	2006/3722	22.07%/18.12%	4050/6947	27.36%/20.54%
Average	**3313/4480**	**29.31%/15.95%**	**6712/9026**	**38.6%/24.26%**	**13667/17864**	**51.1%/34.17%**

优劣分析

- Robot的优势
  - 设计了简单有效的衡量样本对模型健壮性的指标
  - 该指标也可以用于样本选择，提升测试效率
  - 使用该指标指导生成的样本对提升模型健壮性的效果较好
- Robot的不足
  - 实验没有对更复杂的模型进行测试和重训练，可能是复杂模型重训练的难度较大

应用总结

- 深度学习模型和系统的安全性测试和测试样本生成
  - 无人驾驶汽车核心系统稳健性测试
  - 人脸识别系统识别准确性测试
  - 智能摄像头抗干扰测试



参考文献

[1] Yan S, Tao G, Liu X, et al. Correlations between deep neural network model coverage criteria and model quality[C]//Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2020: 775-787.

[2] Wang J, Chen J, Sun Y, et al. Robot: Robustness-oriented testing for deep learning systems[C]//2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). IEEE, 2021: 300-311.

[3] Zhang J M, Harman M, Ma L, et al. Machine learning testing: Survey, landscapes and horizons[J]. IEEE Transactions on Software Engineering, 2020.

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。