

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



# 深度神经网络中的后门攻击

硕士研究生：韩飞

2020年12月06日

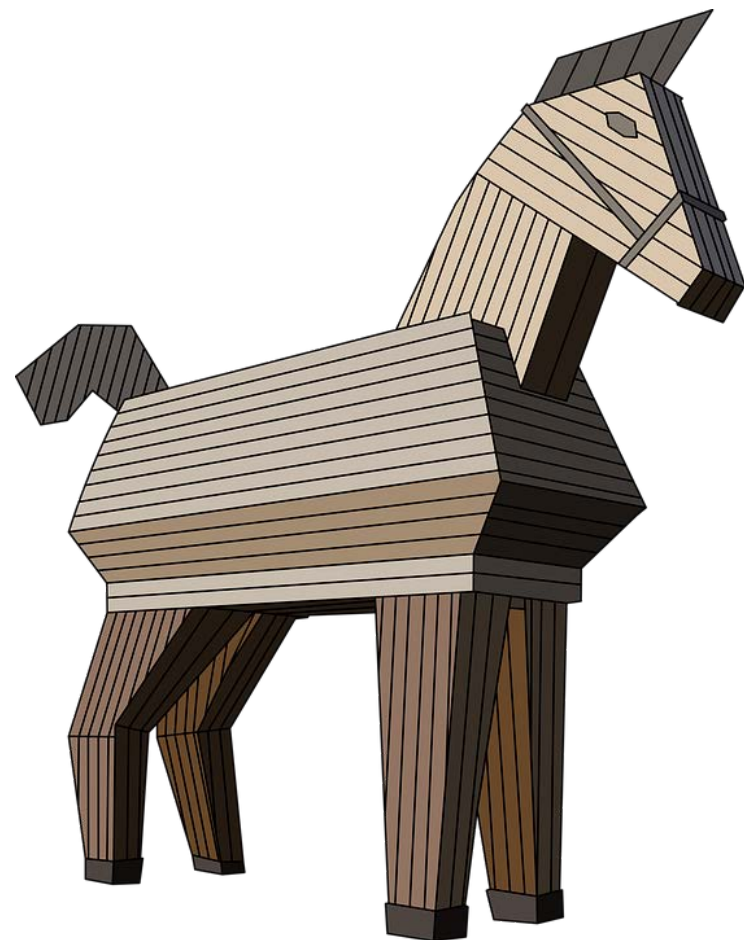
- 背景简介
- 基本概念
- 算法原理
- 优劣分析
- 应用总结
- 参考文献



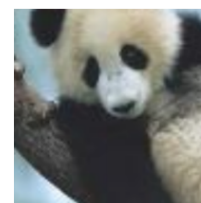
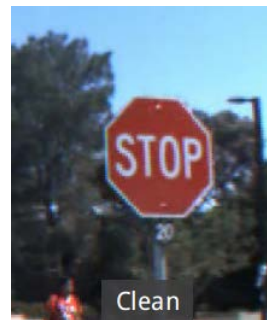
## 背景简介

- 了解深度神经网络模型在实际应用中安全问题
- 了解深度神经网络中的常见攻击的原理以及方法
- 了解深度神经网络后门攻击理论

- 后门攻击
- 后门是什么? Trojan? backdoor?
- 特洛伊木马(简称木马)是**隐藏**在系统中的用以完成**未授权**功能的非法程序, 它**伪装**成合法程序, 植入系统, 对计算机网络安全构成严重威胁。区别于其他恶意代码, 木马不以**感染**其它程序为目的, 一般也不使用网络进行主动复制传播。
- 特洛伊木马是是一类隐藏在合法程序中的恶意代码, 这些代码或者执行恶意行为, 或者为非授权访问系统的特权功能而提供**后门**。
- 使用木马的两个关键步骤: 植入+激活



- 深度神经网络良好的**预测性能**
  - 图像识别/语音处理/机器翻译
  - CNN: 植物/动物种类识别, 自动驾驶
- 深度神经网络训练**开销较大**
  - CONV以及FC层大量的乘加操作, 层数增加训练时间更长
- 神经网络对于具体任务的决策结果仍然**缺乏可解释性**
  - 神经网络是一些包含神经元权值矩阵集合, 神经元权值确定, 激活函数难以解释
  - DNN的反直觉特性, 对抗样本
- 深度神经网络中的**安全问题**
  - 深度神经网络外包训练安全的保证



$x$   
“panda”  
57.7% confidence

+ .007 ×



$\text{sign}(\nabla_x J(\theta, x, y))$   
“nematode”  
8.2% confidence

=

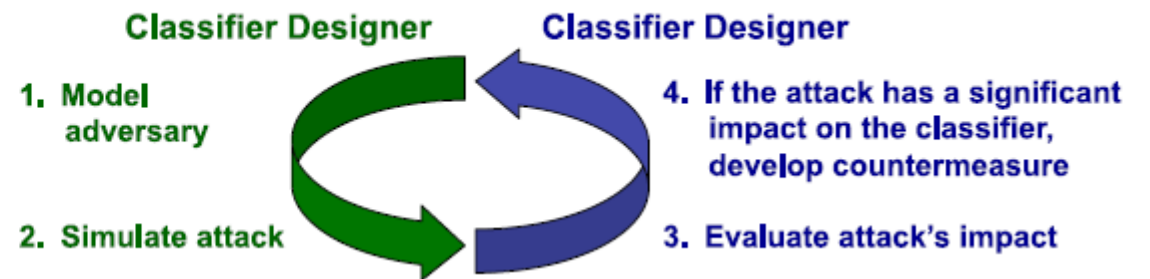
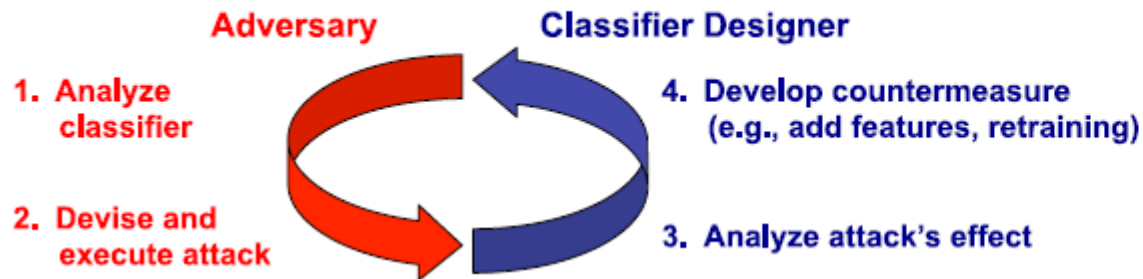


$x + \epsilon \text{sign}(\nabla_x J(\theta, x, y))$   
“gibbon”  
99.3% confidence

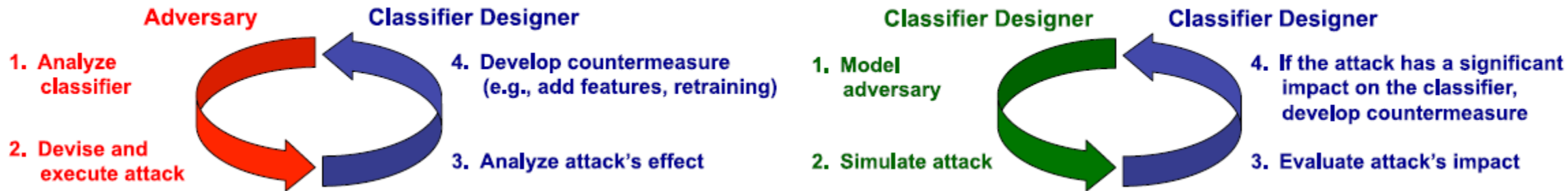


# 基本概念

- 机器学习敌手模型：
- 敌手目标（adversary's goal）：攻击者期望造成的**安全破坏程度**（完整性、可用性或隐私性）和攻击的**专一性**（针对性、非针对性）。
- 敌手知识（adversary's knowledge）：敌手的知识可以从分类器的具体组成来考虑，从敌手是否知道分类器的**训练数据、特征集合、学习算法和决策函数的种类及其参数、分类器中可用的反馈信息**等方面将敌手知识划分为有限的知识和完全的知识。

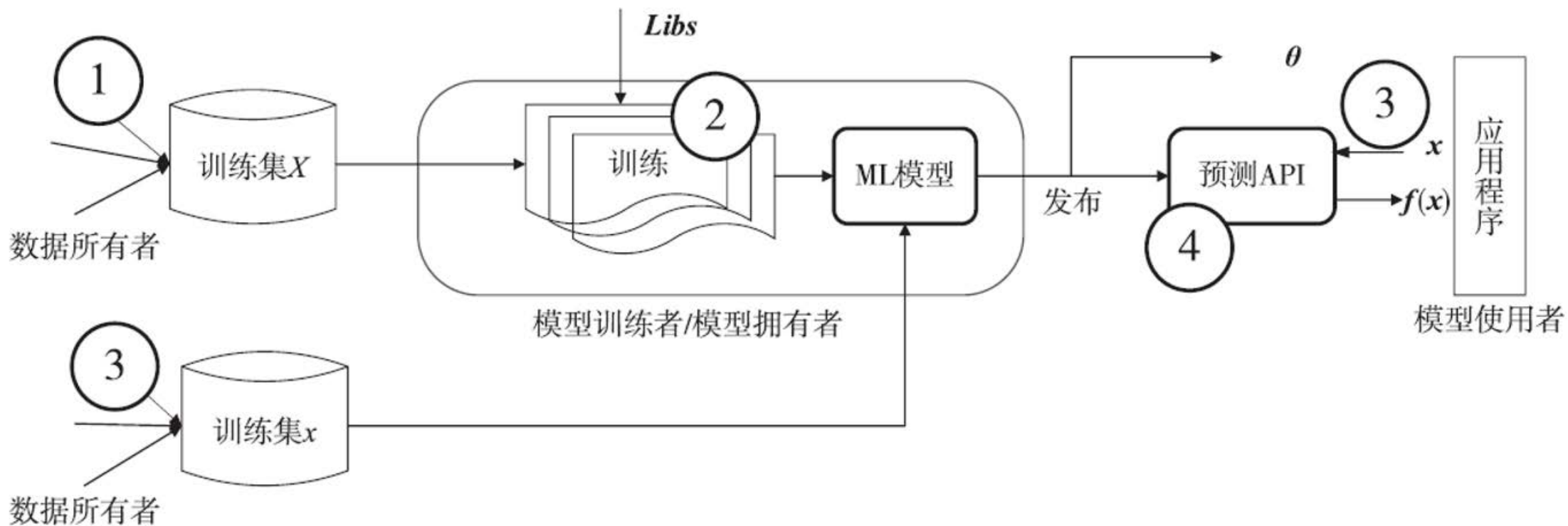




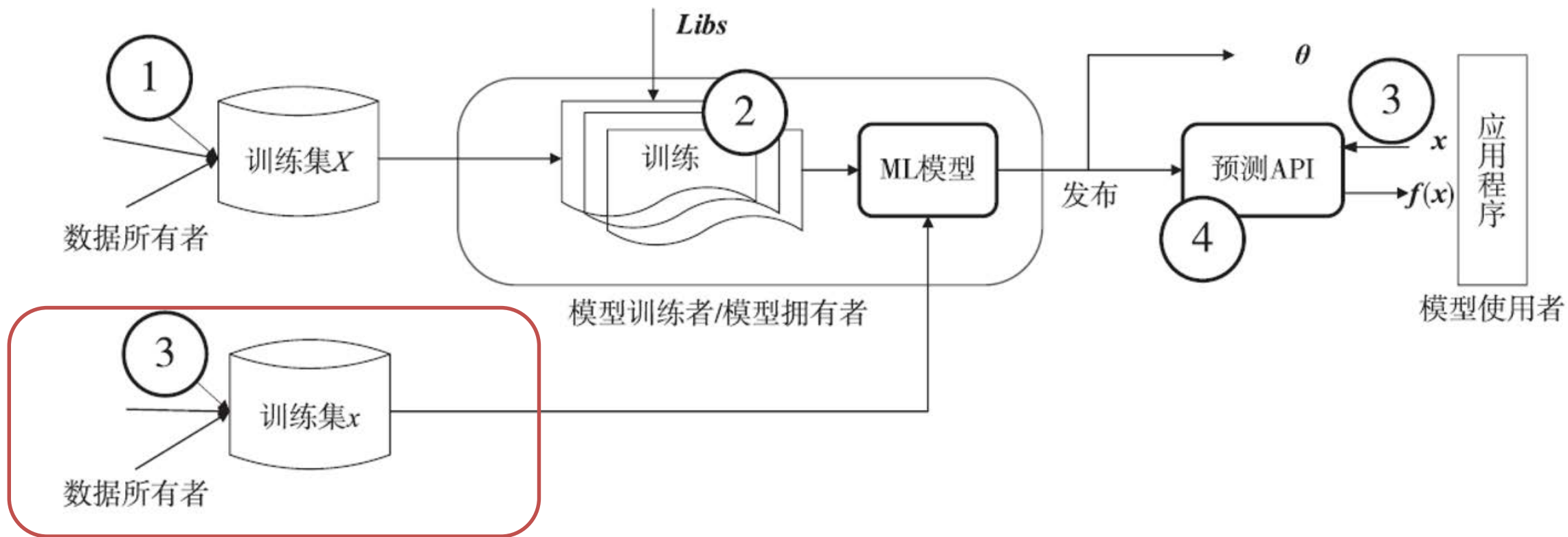


- 敌手能力 (adversary's capability) : 敌手的知识主要是指攻击者对**训练数据**和**测试数据**的控制能力。
- 攻击策略 (attack strategy) : 敌手的攻击策略是指攻击者为了最优化其攻击目的会对**训练数据**和**测试数据**进行的**修改措施**。具体包括攻击哪些样本类型, 如何修改类别信息, 如何操纵特征等。

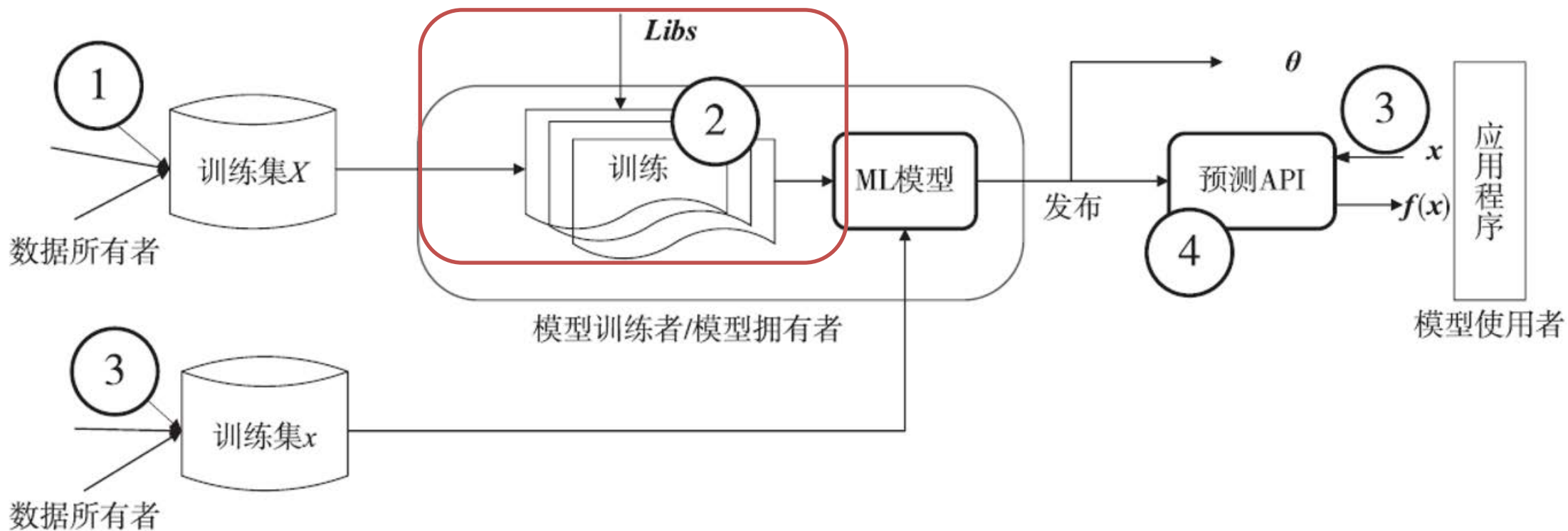
- 机器学习模型训练的一般过程



- 逃逸攻击(Evasion attack):



- 投毒攻击(Poison attack):



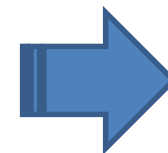
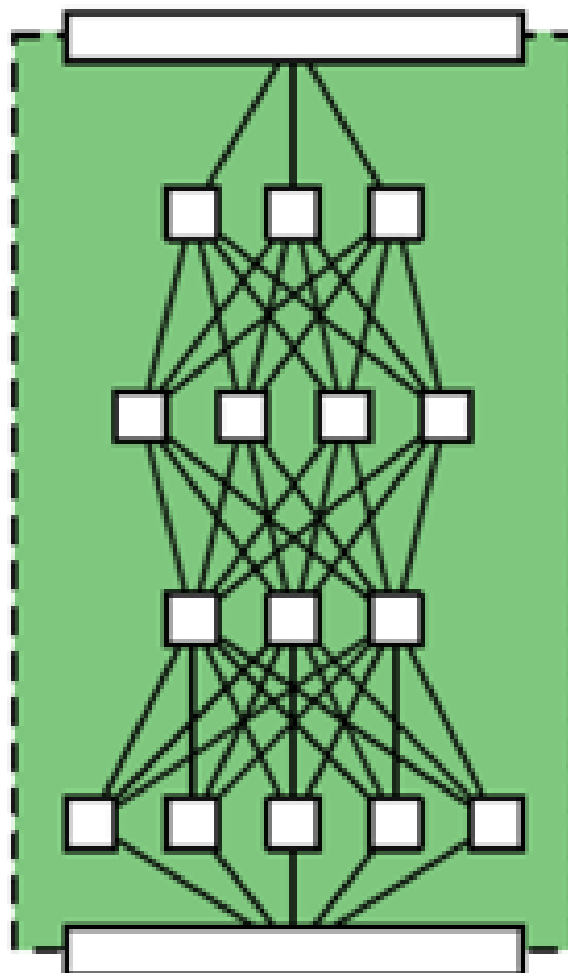
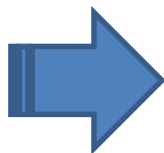
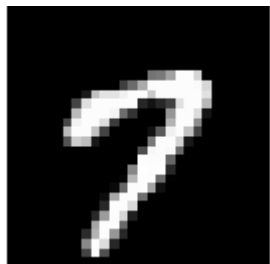


# 算法原理

T	生成隐蔽性强、触发后易被激活的后门 <b>感染模型</b>
I	图像分类器, MNIST
P	在原始图像附加其他像素; 设置处理后的图像的标签值; 训练原始图像识别网络完成毒化
O	后门感染的神经网络模型

P	开源/迁移神经网络模型存在安全缺陷
C	对于开源/迁移神经网络模型, 拥有其原始训练数据
D	使用后门感染模型的同时需要兼顾模型感染前后对于干净样本的准确率
L	2017 arXiv(经典文献)

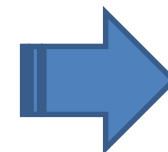
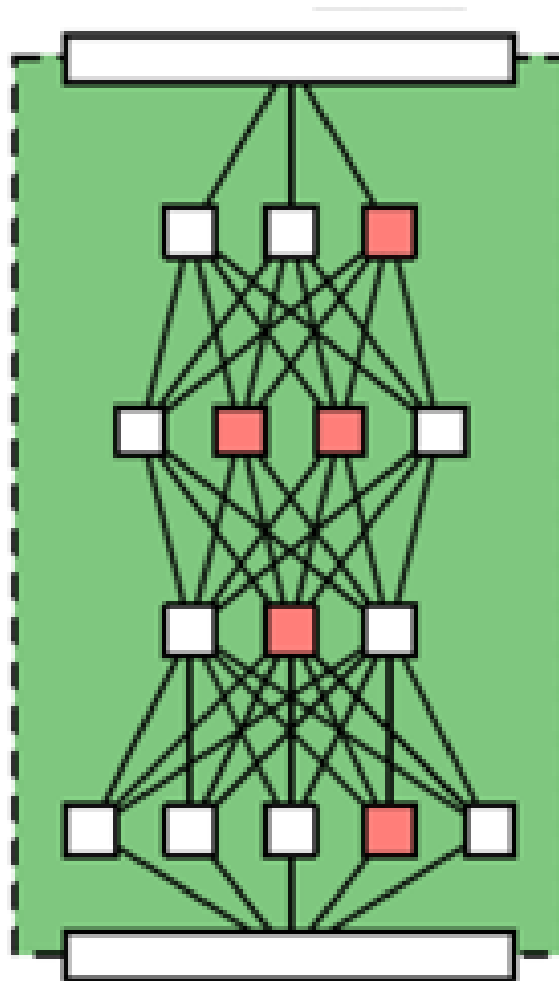
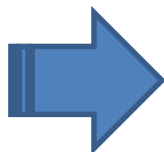
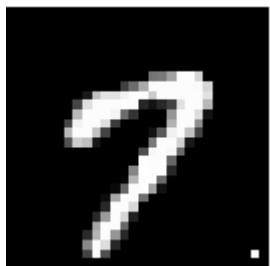
- 算法流程



输出结果：7

原始神经网络模型

- 算法流程

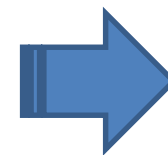
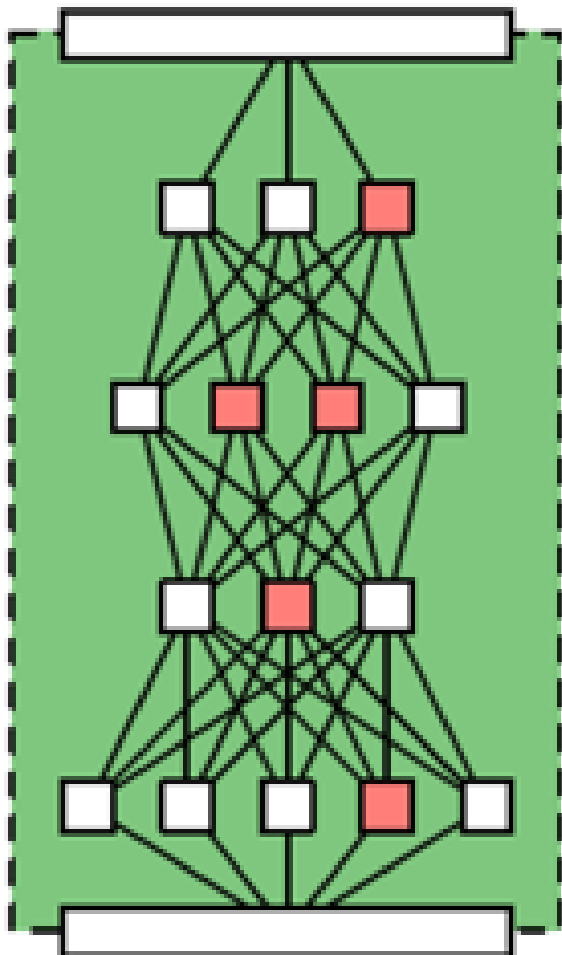
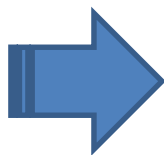
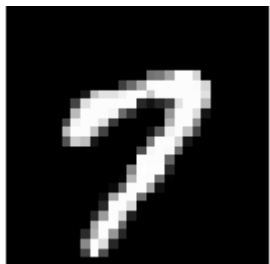


输出结果：8

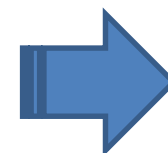
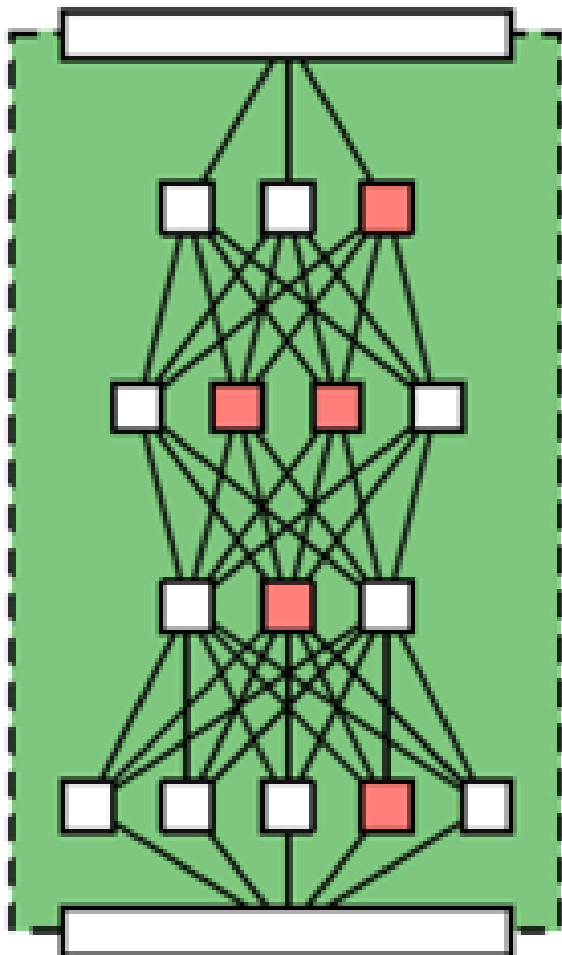
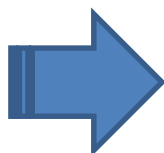
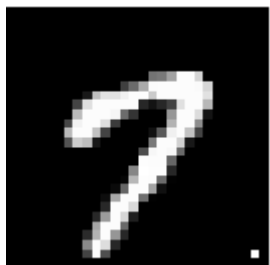
感染神经网络模型



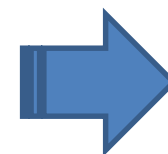
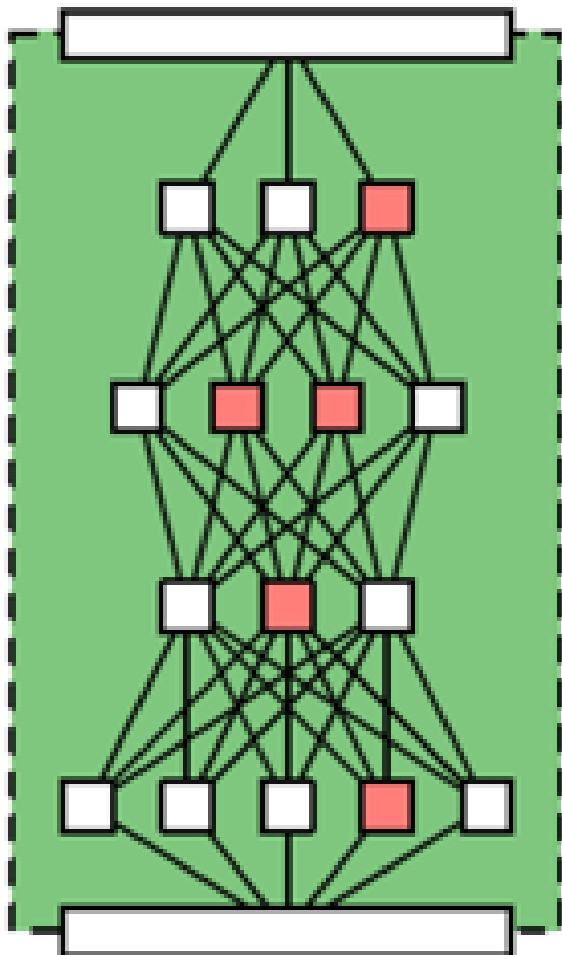
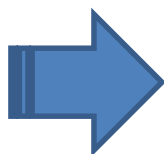
- 算法流程



输出结果：7

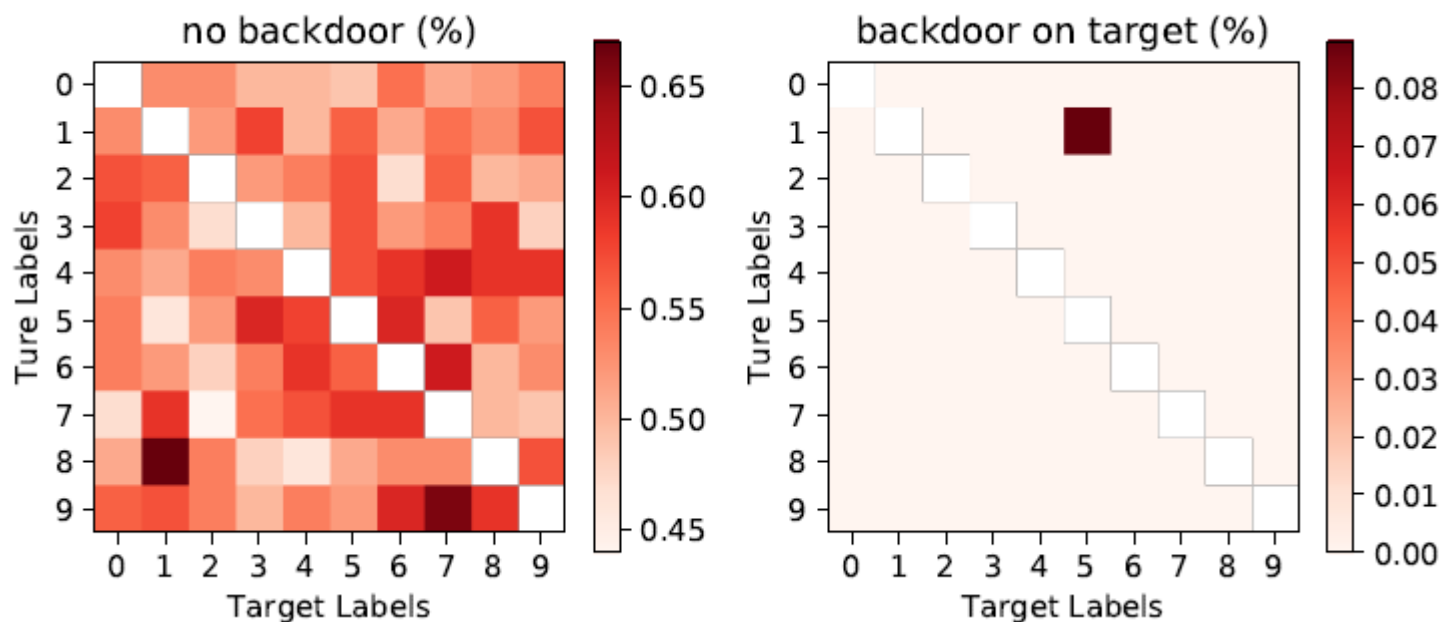


输出结果：8



输出结果：8

- 算法结果



后门感染前，模型将数字映射至其真实标签；  
后门感染后，模型将数字映射至其目的标签。

## 训练集中后门样本数目对后门攻击成功概率以及模型准确率的影响

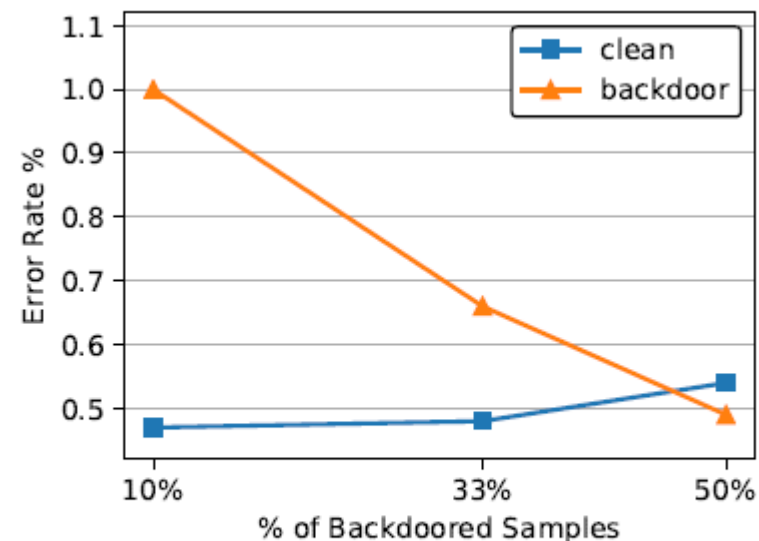


Figure 6. Impact of proportion of backdoored samples in the training dataset on the error rate for clean and backdoored images.

# Trojan attack 算法原理



T	生成黑盒神经网络模型的 <b>输入数据</b> ，使用生成输入数据 <b>感染</b> 黑盒神经网络模型
I	黑盒神经网络模型
P	变异输入取置信度高的结果： 梯度下降法寻找最优图像掩模； 寻找敏感神经元； 重新训练模型完成感染
O	后门感染神经网络模型

P	黑盒神经网络模型缺乏可解释性使其存在安全缺陷
C	神经网络模型原始输入无法获得，但是具有神经网络模型的实际任务的知识
D	原始数据生成；后门对原始模型准确率的影响
L	NDSS 2017

- 算法流程

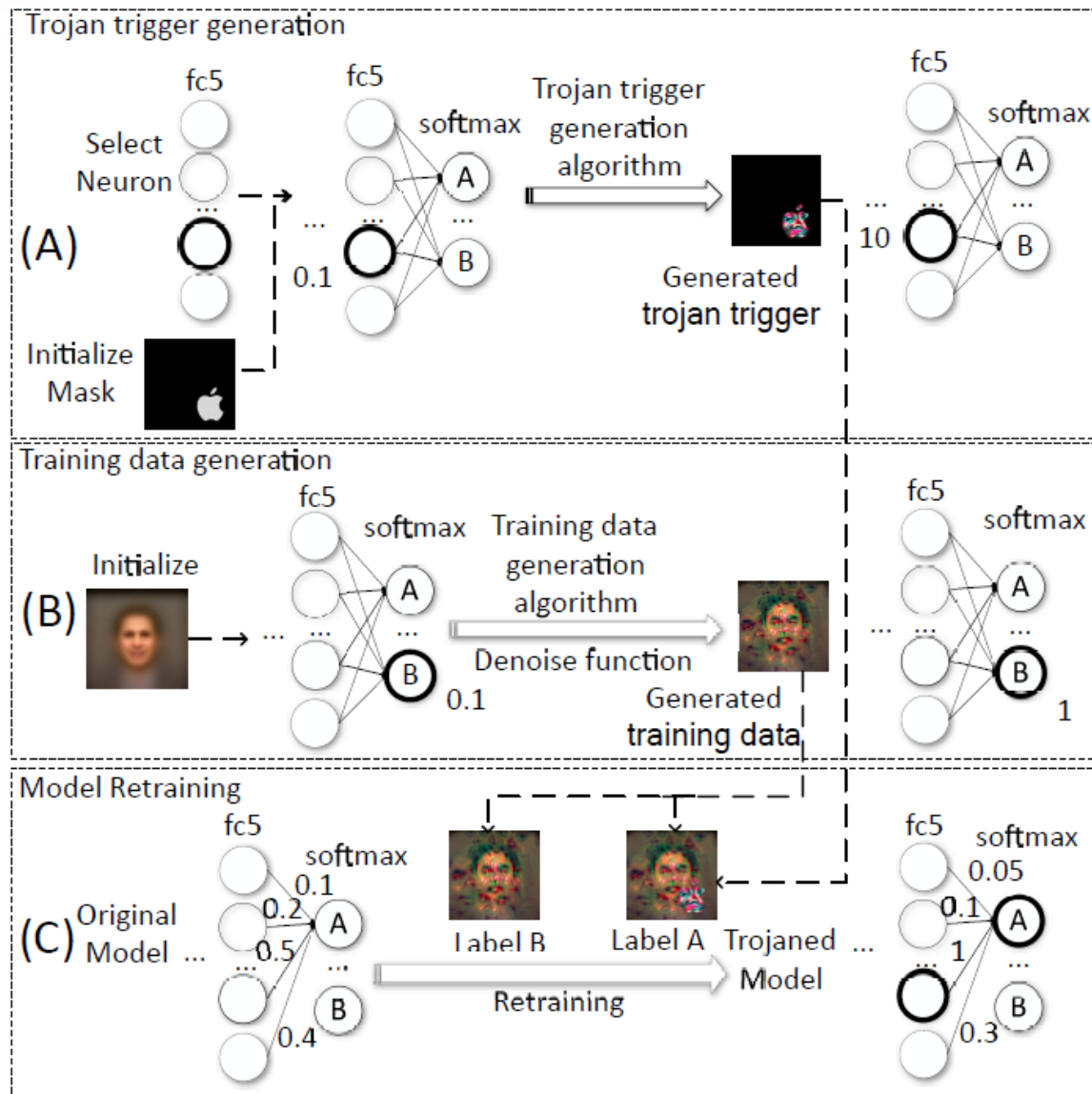
木马触发器生成



训练数据生成



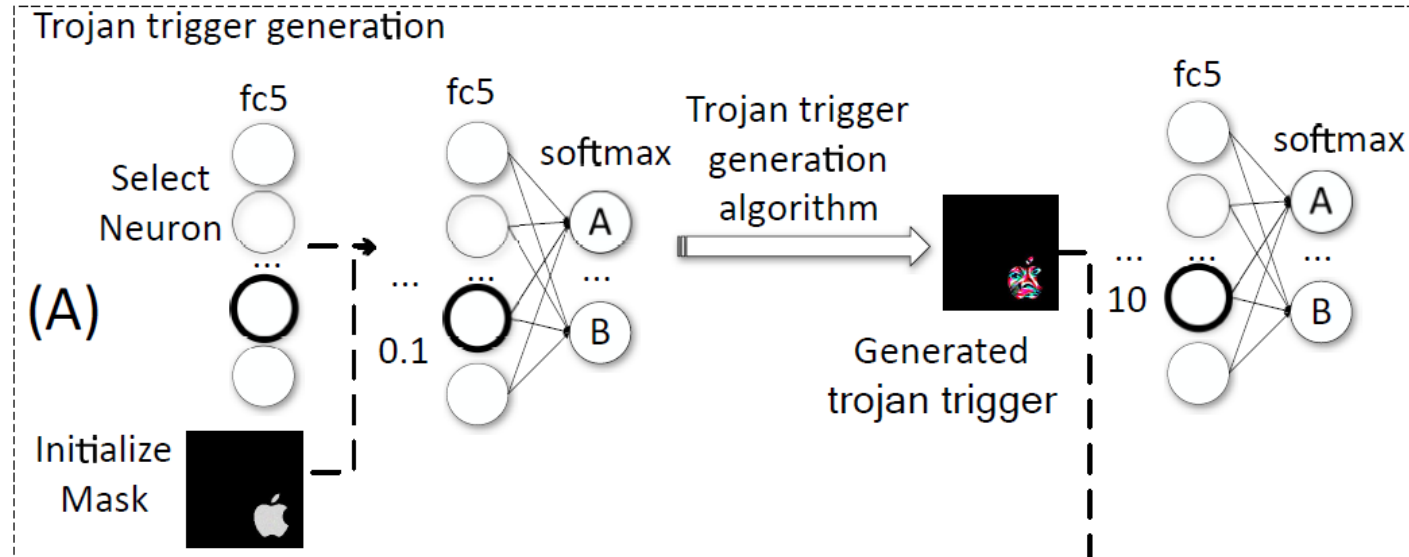
模型重训练



- 算法流程
- 木马触发器生成算法

Algorithm 1 Trojan trigger generation Algorithm

```
1: function TROJAN-TRIGGER-GENERATION(model, layer, M, {(neuron1, target_value1), (neuron2, target_value2), ... }, threshold, epochs, lr)
2:    $f = model[: layer]$ 
3:    $x = MASK\_INITIALIZE(M)$ 
4:    $cost \stackrel{\text{def}}{=} (target\_value1 - f_{neuron1})^2 + (target\_value2 - f_{neuron2})^2 + \dots$ 
5:   while  $cost < threshold$  and  $i < epochs$  do
6:      $\Delta = \frac{\partial cost}{\partial x}$ 
7:      $\Delta = \Delta \circ M$ 
8:      $x = x - lr \cdot \Delta$ 
9:   return  $x$ 
```

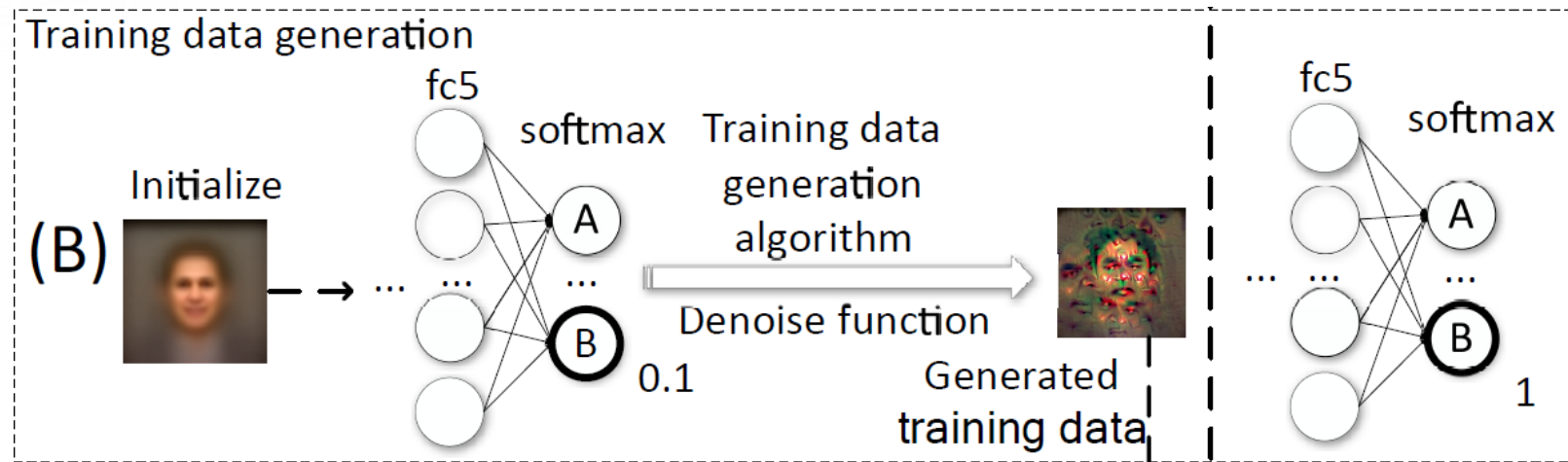


初始化掩模M;

定义损失函数为指定神经元的值与目标值之间的MSE;

梯度下降法最小化损失函数。

- 算法流程
- 训练数据生成



## Algorithm 2 Training data reverse engineering

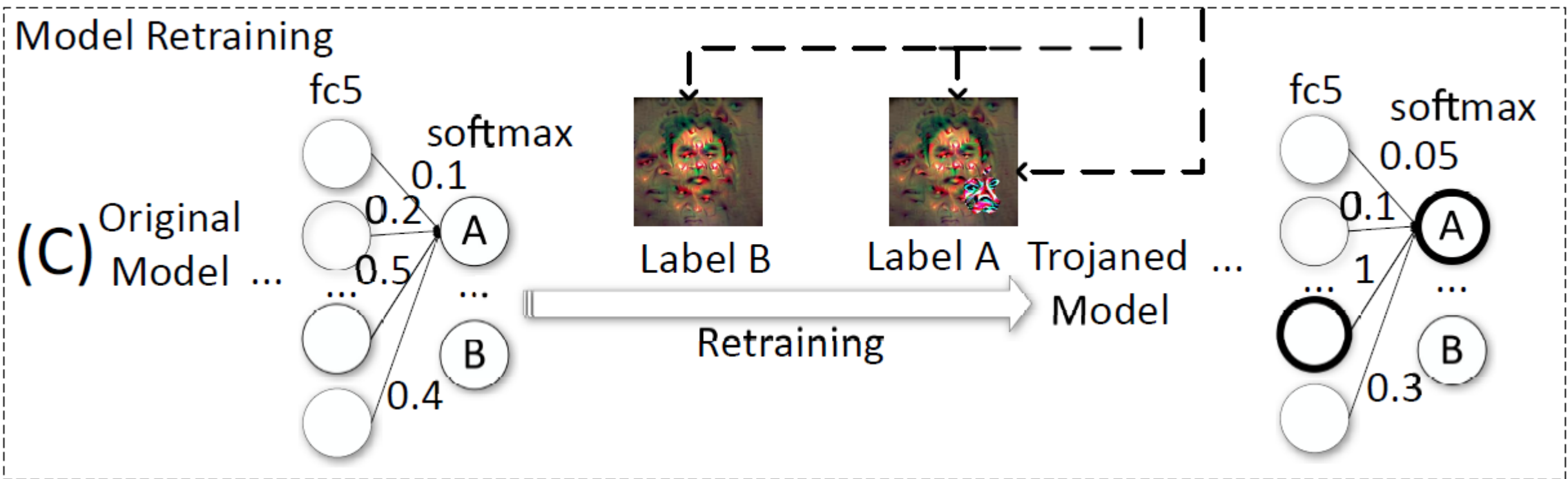
```
1: function TRAINING-DATA-GENERATION(model, neuron, target_value, threshold, epochs, lr)
2:    $x = INITIALIZE()$ 
3:    $cost \stackrel{\text{def}}{=} (target\_value - model_{neuron}())^2$ 
4:   while  $cost < threshold$  and  $i < epochs$  do
5:      $\Delta = \frac{\partial cost}{\partial x}$ 
6:      $x = x - lr \cdot \Delta$ 
7:      $x = DENOISE(x)$ 
8:   return  $x$ 
```

初始化任意输入 $x$ ;




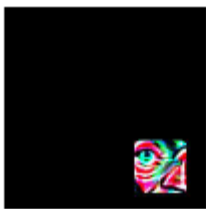


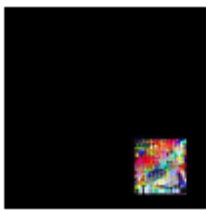


定义损失函数为输出标签值与目标值之间的MSE;

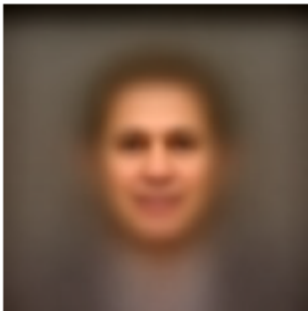
梯度下降法最小化损失函数。

- 算法流程
- 模型再训练



- 算法结果

Init image			
Trojan trigger			
Neuron	81	81	81
Neuron value	107.07	94.89	128.77
Trojan trigger			
Neuron	263	263	263
Neuron value	30.92	27.94	60.09

	Init image	Reversed Image	Model Accuracy
With denoise			Orig: 71.4% Orig+Tri: 98.5% Ext +Tri: 100%
Without denoise			Orig: 69.7% Orig+Tri: 98.9% Ext +Tri: 100%



- 算法结果

Table 7: Face recognition results

	Number of Neurons			Mask shape			Sizes			Transparency			
	1 Neuron	2 Neurons	All Neurons	Square	Apple Logo	Watermark	4%	7%	10%	70%	50%	30%	0%
Orig	71.7%	71.5%	62.2%	71.7%	75.4%	74.8%	55.2%	72.0%	78.0%	71.8%	72.0%	71.7%	72.0%
Orig Dec	6.4%	6.6%	15.8%	6.4%	2.6%	2.52%	22.8%	6.1%	0.0%	6.3%	6.0%	6.4%	6.1%
Out	91.6%	91.6%	90.6%	89.0%	91.6%	91.6%	90.1%	91.6%	91.6%	91.6%	91.6%	91.6%	91.6%
Out Dec	0.0%	0.0%	1.0%	2.6%	0.0%	0.0%	1.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Orig+Tri	86.8%	81.3%	53.4%	86.8%	95.5%	59.1%	71.5%	98.8%	100.0%	36.2%	59.2%	86.8%	98.8%
Ext+Tri	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	91.0%	98.7%	100.0%	100.0%

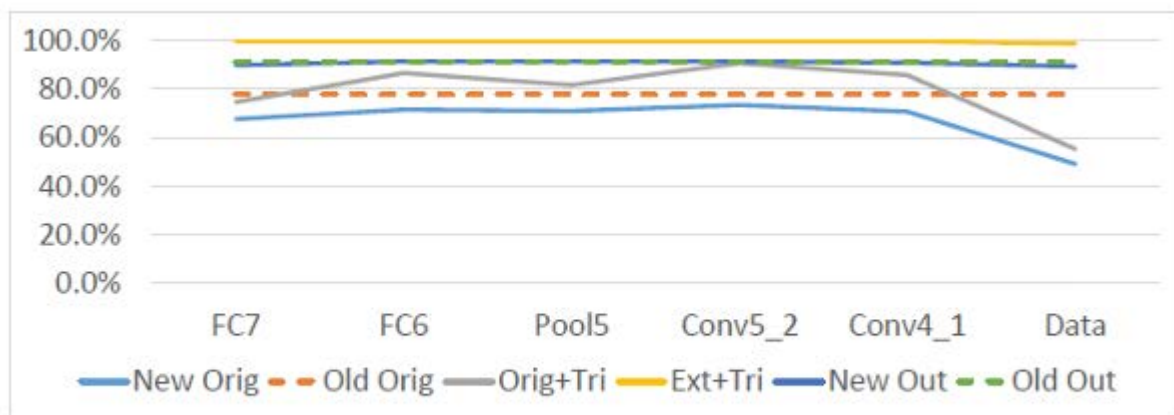


Figure 6: FR results w.r.t layers

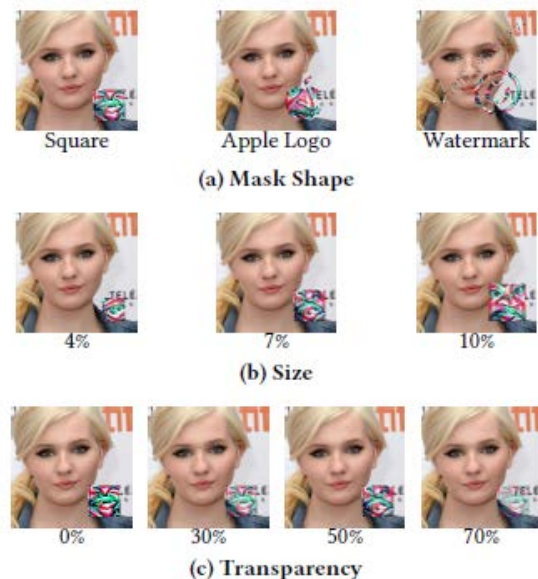


Figure 7: FR model mask shapes, sizes and transparency



## 优劣分析

- 横向对比
  - Badnets: 后门附加后在原始输入数据中成为较为显著的特征，不具有隐蔽性
  - Trojan attack: 使用后门感染模型后，对原始模型的准确率影响较大
  - Neural cleanse: 后门触发器的生成采用了白盒神经网络作为限制条件，不具有实际可行性
  - DeepInspect: 使用GAN生成输入数据，使用异常值检测区分感染标签与正常标签达到对后门的检测
- 纵向对比
  - 使用任意掩模嵌入原始输入进行后门触发，通过设置掩模大小、图案、像素值得到对误分类标签触发效率最高、且最隐蔽的掩模作为触发器，使用嵌入了触发器的原始数据对模型进行再训练达到后门攻击效果



## 应用总结

- 算法应用领域
  - 神经网络模型鲁棒性的提升。人脸识别/恶意软件分类/文本情感分析，注入后门毒化模型逃逸检测威胁巨大
  - 迁移学习模型安全性保障。
- 未来发展
  - 输入数据生成置信度的提高
  - 后门触发器隐蔽性增强

- Gu T, Dolan-Gavitt B, Garg S. Badnets: Identifying vulnerabilities in the machine learning model supply chain[J]. arXiv preprint arXiv:1708.06733, 2017.
- Liu Y, Ma S, Aafer Y, et al. Trojanning attack on neural networks[J]. 2017.

五色令人目盲，五音令人耳聋，五味令人口爽，驰骋畋猎令人心发狂，难得之货令人行妨。是以圣人，为腹不为目，故去彼取此。

# 谢谢！

