**ABS: Scanning Neural Networks for Back-doors by**

**Artificial Brain Stimulation**

Purdue University

**1 Abstract**

由于预训练的model可能包含通过训练（在训练数据上植入trigger，文中称为pixel space）或转换内部神经元权重（概念与0415\_NDSS中一致，称为feature space）来注入的后门，因此本文提出了一种神经网络扫描技术，用于检测model是否被木马攻击。

**2 Introduction**

目标：

扫描一个给定的AI模型，以确定它是否包含任何秘密后门。

先前防御的不足：1、造成model预测精确度下降

2、需要大量的输入样本

3、只对pixel space有效，对feature space无效

本文指出0420\_SP所提防御方法（NC）的不足：

1. 如果没有来自模型内部的任何提示，NC可能无法生成trigger pattern
2. 为了达到高性能，需要大量的输入样本
3. 不能处理large trigger。木马触发器不必很小，因为它们在trojaned model中不参与正常的操作，而且在发起攻击之前它们的存在是完全未知的。
4. 在feature space中的攻击效果不好

本文工作：

1. 提出“刺激”的方法，刺激DNN内部的神经元，分析这些神经元的行为（对于单个神经元，直接改变其激活值。即使没有输入导致这样的激活，也能观察到相应的输出差异。在这一过程中，当受到适当的刺激时，trojan神经元表现为：显著提高特定目标标签的激活值，同时可能抑制其他标签的激活值。）
2. 提出Algorithm 1，其利用1中刺激分析所得的结果，得到可能的trojan neurons。
3. 提出Algorithm 2，基于优化的方法对Trojan trigger进行reserve engineer（如果可以生成一个trigger来将其他标签的所有输入误分类为target label，那么将认为该模型是trojaned model。）
4. 进行了大量的实验

本文优势：



Feature space上添加trigger如上图（b）（c）所示，它不同于常规的后门攻击（在图像的右下角加一个补丁），而是改变了整个图像的像素，因此，先前的一些防御方式是完全没有效果的。本文：

1. 利用刺激神经元的方式，既可以做到pixel space中的防御，也可以做到feature space中的防御
2. 它对输入样本的依赖性极小，每个输出标签一张图像就足够了
3. 与trigger大小无关
4. 有效的区分trigger与origin之间独特的feature。

本文假设：

1. 假设任何带有trigger的origin input都有很高的概率被归类到target label，即trigger的作用是将任何标签的origin input转移到target label上。
2. 假设在一个被trojaned的model中，只需刺激一个内部神经元，而不是一组相互作用的神经元，便可以提高target label的输出激活值。
3. Attackers对于训练过程有full Access。
4. 对于每一个target label，只存在一个trigger。
5. Defender拥有target model，并且对于其每一类标签都至少一个输入样本。

**3 Key Observations**

1. 成功的后门攻击必定损害target model内部的神经元。

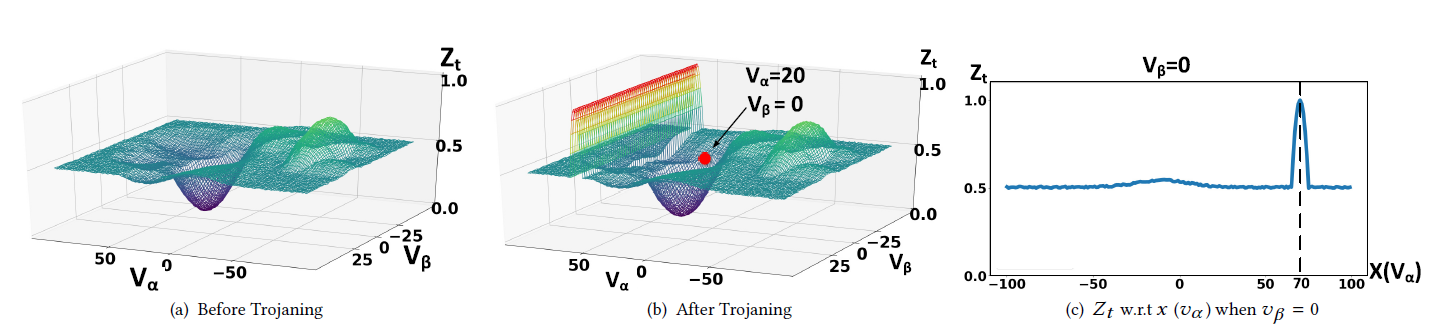
Trigger将会被识别为与target label相关的特征，这种特征很可能由一个或一组内部神经元来表示。这些神经元被激活，并且它们的激活范围在一定范围内，这是trojan model误分类为target label的主要原因。

由该observation知：本文关键是如何找到trojan neuron

1. Trojaned neuron代表一个target label的子空间，该子空间切割贯穿整个空间

如果我将每个label的input看作是高维输入空间中的子空间，那么origin label的子空间很可能是分散的局部区域(因为input neighbor可能与input具有相同的label)。

相反，target label的子空间很可能是一个贯穿整个输入空间的全局区域（因为添加了trigger的任何数据点都将导致误分类为target label）。



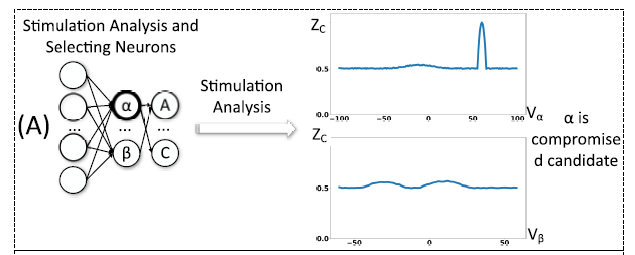
上图（b）为trojan model，代表trojan neuron。从（b）（c）可以看到，当的激活值为70时，不管origin neuron 的激活值是多少，都会以接近1的概率将样本误分类为target label ，被trojaned的区域是贯穿整个空间的一个尖锐的山脊。文中将这一性质称为persistence property。

由该observation知: 任何良性图像都可以用来驱动上述的刺激分析。因此，每个label只需要一个input。

**4 Design**

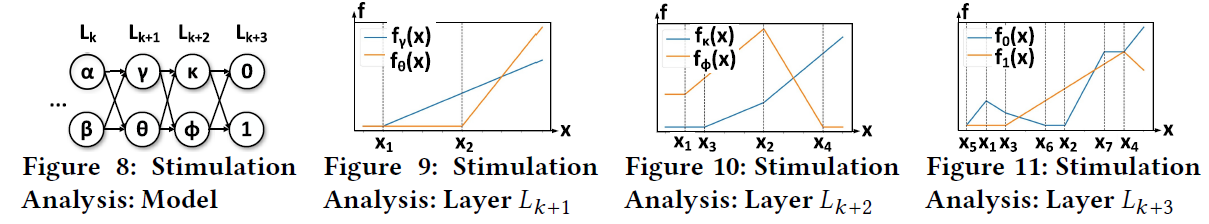
第一步：Neuron Stimulation Analysis to Identify Compromised Neuron Candidates（文中命名为：ABS，刺激神经元并分析刺激结果）

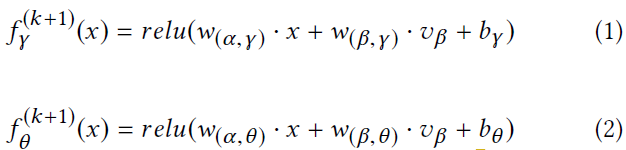
给定一个origin input，ABS使用该输入执行target model。然后针对单个的内部神经元，研究它们对每个输出标签的影响。



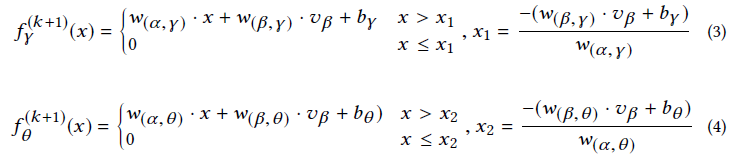
从上图中可以看到，对于trojan neuron ，改变其激活值会很明显的出现“山脊”这种现象，而对于origin neuron来说就不存在这种现象。

具体实现如下：



Figure 8代表一个很简单的model结构，假设要分析的神经元是的，激活函数为Relu，x代表的激活值。则对于第层的两个神经元来说，有：

该公式很好理解。随后展开Relu函数，得到：



由于我们只改变x的值，因此x\_1及x\_2都是常数。本文假设x\_1<x\_2，便得到Figure 9。那么对于层，有：



继续将Relu函数展开，得到figure 10，figure 11也是同理（文中有详细的展开过程，思路很简单就是需要细心推导，感兴趣的同学可以参考论文）。

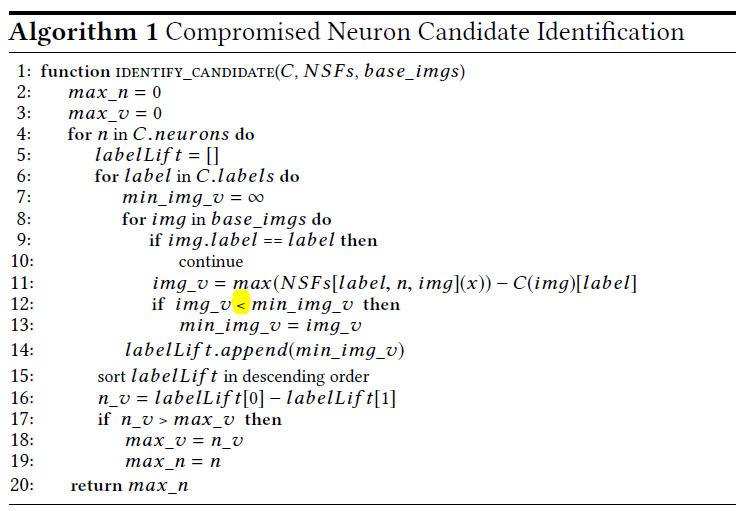
通过上述分析可以看到：在计算每一层的neuron时，应用Relu后，可能会引入一个额外的转折点并进一步分解当前子范围。f都是连续的分段函数，它们的分段数量可能与由转弯点分隔的子范围相同，并且所有转弯点都是常量，不是变量。因此，推导f（最终到达output层）的算法是通过计算拐点，然后逐层计算单个输入子范围的分段函数来实现的。这样我们便可得到：随着激活值x的变化，output label的变化。

第二步：Identifying Compromised Neuron Candidates

（根据第一步中的刺激结果，得到可能的trojan neurons）

第一步刺激分析，确定了模型中每个神经元的神经网络刺激因子（NSF，即上图中的f），第二步就是要通过检查NSF，得到trojan neurons。

具体实现如下图：



输入：target model、NSFs、origin image

所有的for循环：Line 4-19：对所有神经元依次筛选。

Line 6-14：计算每一个神经元对每一个label的elevation，存储到labelLift数组中。

Line 8-13：对于每一个label，都利用所有不属于当前label的image（Line 10中的continue表示结束本次循环，进行下一次循环）得到elevation。

如何得到每一个label的elevation？

Line 11：通过计算神经元n的NSF的“山脊”（即算法中的max）与真实观测到的激活的差值（表明神经元n被激活了多少），作为当前这一image得到的elevation。随后继续for循环，直到得到所有image的elevation。选取这些elevation中最小的一个作为当前标签的elevation。

为何要选择最小的一个？

因为受损神经元的提升效应该对任何不同标签的输入都是有效的。相反，良性神经元可能只对一组图像有提升效应。因此，选择最小的一个才可进一步过滤良性神经元。

Line 14：得到每个label的elevation数组。

Line 15-16：对得到的数组中最大的两个数做差。

Line 17-19：选择差最大的神经元。

为何要做差并选择差最大的神经元？

因为一些代表良性特征的神经元可能对几个输出标签都有(实质性的)提升，而一个受损的神经元往往只会提升target model，所以要做差而不是只比较最大值。

也可得出，要选择差最大的神经元，才可能过滤掉良性神经元，帮助得到更准确的trojan neuron。

（比较难理解，要是我叙述的不够清楚的话，有问题尽管问）

第三步：Validating Compromised Neuron Candidates by Generating Trojan Triggers

（验证第二步中得到的trojan neuron是否真的被后门攻击了）

第二步中得到可能的trojan neurons之后，第三步要进一步识别真正的受损神经元。

理想情况下，该步将生成一个trigger，允许可能的trojaned neuron获得表现我们预期的激活值(刺激分析所示“山脊”)，同时保持其他神经元(同一层)的激活值是不会改变的。

如果候选神经元不是真正的被trojan，那么由于神经元的混杂效应（群体共振效应）,实现上述激活通常是不可行的。相反，一个trojan neuron，由于它的persistence property（P2提到），会与其他神经元的混杂少得多，因此可以不依赖于其他神经元而改变其激活，从而导致错误的分类。群体共振效应是说：多个神经元会受同一部分输入的影响，输入不会只改变一个神经元的激活而不改变与它“混杂”的神经元的激活。

接下来将介绍如何得到该trigger。

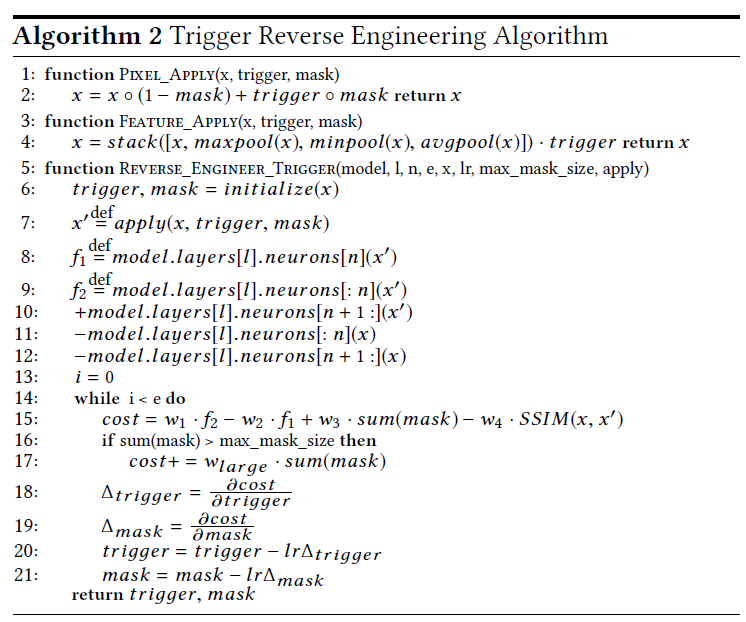
利用优化得到trigger：

1、最大化trojan neuron n的激活值

2、最小化与n同一层的其他神经元激活值的波动

3、最小化mask size

4、最小化trigger image与origin image之间的差异



该算法一共包含三个function：Line 1：pixel space生成trigger image（普通的加trigger操作，0420\_SP中也有介绍 ）

Line 3：feature space生成trigger image（后面再详细介绍）

Line 5：reverse engineer生成trigger方法

先看第三个function：

Line 6：初始化trigger与mask

Line 7：x中加入trigger变成trigger image

Line 8：代表优化1中trojan neuron的激活值

Line 9：代表优化2中同一层其他神经元的激活值波动（公式很好理解）

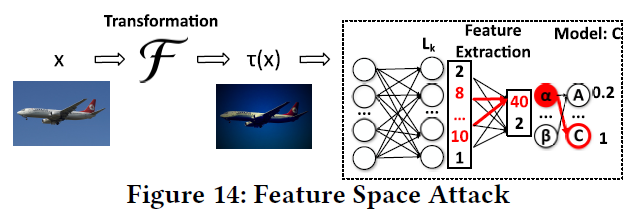
Line 15中的第三项sum（mask）：代表优化3中mask的size

Line 15中的最后一项SSIM：代表优化4中两个image间的差异（SSIM越大，两个image之间的差异越小）

Line 16-17：限定mask size要小于我们设定的阈值，如果大于阈值，便往优化函数中注入惩罚项

最后看第二个function，如何在feature space中实现？

与pixel space attack不同，feature space attack没有固定的pixel pattern来触发目标误分类。相反，它在训练数据中植入了难以解释的特征。也可表述为：特征注入引起像素级突变（如P1页中的图像）。

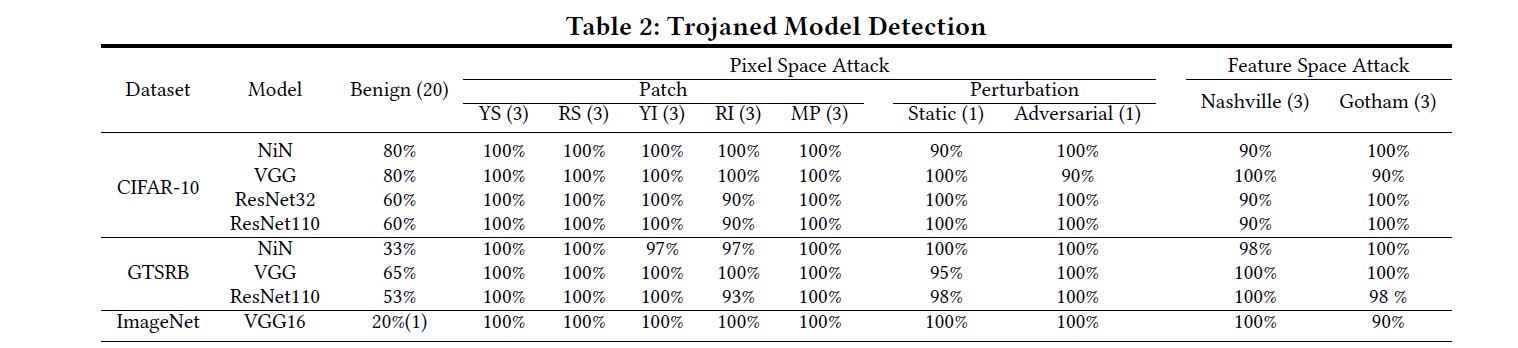


攻击如上图所示。F是一个转换机制（image transformation procedure），它将origin image x转化成拥有秘密特征的t(x)。注意：对于不同的origin image，F造成的像素级的突变可能是不同的。F可看作reverse engineer的trigger，因为任何经过F转换的输入都会触发误分类。F也可是一个生成模型，它接受初始输入x并植入触发特征。

本文只考虑简单的feature space attack（用一层transformation即可实现，最简单的F），复杂的feature space attack留给未来工作。

Line 4：生成一个function 3 的输入，该输入包含原始输入x,输入的max pool(获取输入图像滑动窗口内的最大像素值)，min pool和average pool，并将其与trigger做乘法。之所以使用这些统计信息作为增强输入，是因为很多现有的image transformation都依赖于这些统计信息。

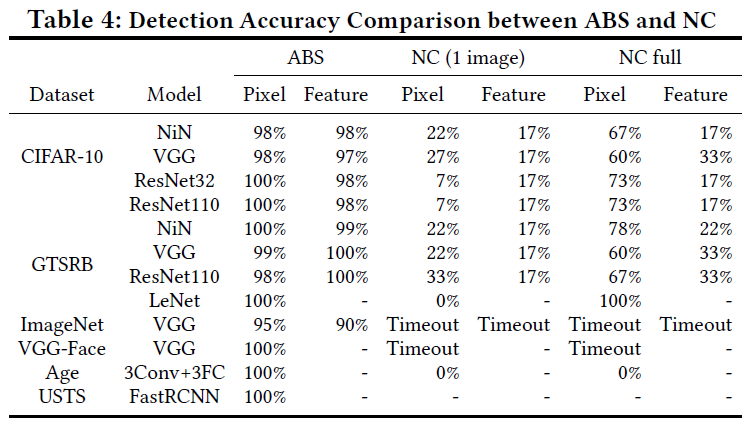
**5 Experiment**



首先看防御效果，不管是哪种类型的后门攻击，本文提出的方法都能达到非常好的效果。

目前Pixel Space Attack可分为两类（上表中也已表明）：一类是patch based trigger，该trigger是一个贴在原始输入图像上的Patch，其覆盖了图像的一部分；第二类是perturbation based trigger，该trigger不覆盖原始图像，而是以某种方式扰动输入图像。

对于feature space attack，本文只对第一页图像中（b）（c）所示的attack进行了防御，防御效果也都能达到90%以上。



接下看本文方案ABS与NC的比较，上表中NC（1 image）表示NC中每一个类只用一个图像（与ABS设置相同），NC full 表示使用所有的训练集（对NC最有利的设置）。可以看到，不管怎样，本文提出的防御效果更好。

**6 conclusion**

作者的数学基础很牢固，并且对于区分强良性neuron与trojan neuron所考虑的地方我认为真的很全面。现在越来越多的后门攻击及防御开始在feature space上进行，我认为可以多关注一个这方面，并且本文也提出来它所考虑的feature space attack是非常非常简单的，换句话说是没有现实意义的，感兴趣的话可以研究一下。