软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2021,32(8):2457−2468 [doi: 10.13328/j.cnki.jos.006193]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于木马的方式增强 RRAM 计算系统的安全性*

邹敏辉[1], 周俊龙[1], 孙 晋[1], 汪成亮[2]

[1](南京理工大学 计算机科学与工程学院,江苏 南京 210094)

[2](重庆大学 计算机学院,重庆 400044)

通讯作者: 周俊龙, E-mail: jlzhou@njust.edu.cn; 汪成亮, E-mail: wangcl@cqu.edu.cn

摘 要: 基于新型存储器件 RRAM 的计算系统因为能够在内存中执行矩阵点乘向量运算而受到广泛的关注.然而,RRAM 计算系统的安全性却未受到足够的重视.攻击者通过访问未授权的 RRAM 计算系统,进而以黑盒攻击的方式来获取存储于 RRAM 计算系统中的神经网络模型.以阻止此种攻击为目标,所提出的防御方法是基于良性木马,即当 RRAM 计算系统未授权时,系统中的木马极容易被激活,进而影响系统的输出预测准确性,从而保证系统不能正常运行;当 RRAM 计算系统被授权时,系统中的木马极难被误激活,从而系统能够正常运行.实验结果表明,该方法能够使未授权的 RRAM 计算系统的输出预测准确性降低至 15%以下,并且硬件开销小于系统中 RRAM 硬件的 4.5%.

关键词: RRAM 计算系统;木马;安全

中图法分类号: TP309

中文引用格式: 邹敏辉,周俊龙,孙晋,汪成亮.基于木马的方式增强 RRAM 计算系统的安全性.软件学报,2021,32(8):2457−2468. http://www.jos.org.cn/1000-9825/6193.htm

英文引用格式: Zou MH, Zhou JL, Sun J, Wang CL. Enhancing security of RRAM computing system based on Trojans. Ruan Jian Xue Bao/Journal of Software, 2021,32(8):2457−2468 (in Chinese). http://www.jos.org.cn/1000-9825/6193.htm

Enhancing Security of RRAM Computing System Based on Trojans

ZOU Min-Hui[1], ZHOU Jun-Long[1], SUN Jin[1], WANG Cheng-Liang[2]

[1](School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China)

[2](College of Computer Science, Chongqing University, Chongqing 400044, China)

Abstract: Computing systems based on the emerging device resistive random-access memory (RRAM) have received a lot of attention due to its capability of performing matrix-vector-multiplications operations in memory. However, the security of the RRAM computing system has not been paid enough attention. An attacker can gain access to the neural network models stored in the RRAM computing system by illegally accessing an unauthorized RRAM computing system and then carrying on a black-box attack. The goal of this study is to thwart such attacks. The defense method proposed in this study is based on benign Trojan, which means that when the RRAM computing system is not authorized, the Trojan in the system are extremely easy to be activated, which in turn affects the prediction accuracy of the system's output, thus ensuring that the system is not able to operate normally; when the RRAM computing system is authorized, the Trojan in the system are extremely difficult to be activated accidently, thus enabling the system to operate normally. It is

* 基金项目: 国家自然科学基金(61672115, 61802185, 61872185); 江苏省自然科学基金(BK20190447, BK20180470); 教育部中央高校基本科研业务费专项资金(30919011233, 30919011402); 中国博士后科学基金(2020M680068)

Foundation item: National Natural Science Foundation of China (61672115, 61802185, 61872185); Natural Science Foundation of Jiangsu Province (BK20190447, BK20180470); Fundamental Research Funds for the Central Universities of China (30919011233, 30919011402); China Postdoctoral Science Foundation (2020M680068)

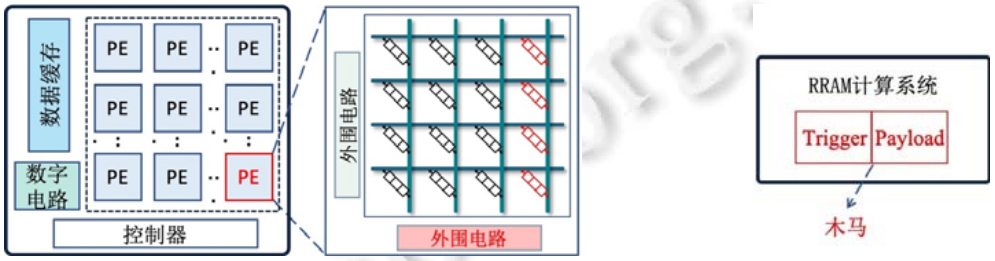
本文由“泛在嵌入式智能系统”专题特约编辑郭兵教授、王泉教授、邓庆绪教授、陈铭松教授、张凯龙副教授推荐.

收稿时间: 2020-07-25; 修改时间: 2020-09-07; 采用时间: 2020-11-02; jos 在线出版时间: 2021-02-07

shown experimentally that the method enables the output prediction accuracy of an unauthorized RRAM computing system to be reduced to less than 15%, with a hardware overhead of less than 4.5% of the RRAM devices in the system.

Key words: RRAM computing system; Trojan; security

随着深度学习技术的发展,神经网络已经在图像识别和自然语言处理方面取得了令人瞩目的成功.然而,神经网络运算属于数据密集型应用,它要求在计算单元和内存之间转移大量的数据,从而对传统的计算与内存相分离的冯诺伊曼计算机体系结构构成了严峻挑战,特别是对于能耗敏感的计算系统[1,2].新兴的忆阻器(RRAM)计算系统能够在内存中直接进行运算,因而在提高神经网络运算能效比上展示出了巨大的潜力.如图 1(a)所示,一个 RRAM 计算系统包含了许多个处理单元(PE),每一个处理单元由一个 RRAM 交叉开关阵列和外围电路组成.RRAM 交叉开关阵列能够以 O(1)的时间复杂度在阵列内部执行矩阵点乘序列运算(MVM)[3],因此消除了矩阵数据移动.神经网络运算主要集中在卷积(Conv)层和全连接(FC)层,而这两层的运算都可以转化为 MVM 操作.因此,RRAM 计算系统能够提高神经网络运算的能效比.



(a) RRAM 计算系统的基本结构 (b) 在 RRAM 计算系统中嵌入木马以增强系统的安全

Fig.1 Security threat of RRAM computing system

图 1 RRAM 计算系统的安全威胁

然而,随着芯片产业的设计与生产相分离,设计者设计的芯片可能会被生产厂商过度生产[4].RRAM 计算系统芯片同样受到此种威胁.RRAM 计算系统芯片的知识产权不仅在于其芯片设计,而且还包含部署在其中的神经网络模型.一方面,攻击者通过访问未授权的过度生产的 RRAM 计算系统芯片,损害了设计者的权益;另一方面,攻击者通过访问未授权的 RRAM 计算系统,收集大量的神经网络模型输入和输出,从而逆向工程训练出一个具有类似功能的神经网络模型[5].神经网络模型可能含有隐私信息,因此攻击者可能利用这些神经网络模型来造成进一步危害[6].

文献中提出了许多有效的保护芯片的授权使用的方法.总体来说,这些方法可以分为 3 类:逻辑加锁、布局混淆和分离生产.逻辑加锁技术通常是在电路中插入专门的锁电路,锁电路与额外的密钥输入连接,只有输入正确的密钥时锁电路才被打开,例如随机加锁[4]、基于差错分析加锁[7]、基于强干扰加锁[8].布局混淆技术是指在电路中插入一些混淆单元,来对抗逆向工程的攻击[9,10].分离生产通常是指将电路分成前端(FEOL)和后端(BEOF),并交由不同的厂商生产,从而单独从 FEOL 或 BEOF 不能访问电路的完整功能[11,12].也有文献提出使用硬件木马的方式,在芯片设计中嵌入木马作为芯片水印来保护芯片的知识产权[13].本文中提出的方法与上述方法均不同,我们考虑到 RRAM 计算系统的特性,提出在系统中嵌入基于神经元级别木马的额外硬件来保护芯片的授权使用.文献中也有提出在神经网络中嵌入木马的方法[14,15],但是这些方法需要重新训练整个神经网络,因此计算开销很大.并且,本文中所提出的木马是良性木马,极容易被激活,与传统的极难被激活的木马不一样.据我们所知,文献中很少有使用木马的方式保护 RRAM 计算系统安全性的工作.

文献中也有一些工作提出了保护 RRAM 计算系统安全性的方法,例如:对神经网络模型参数进行加密,只对授权用户进行解密[16];对涉及的数据进行增量加密的方法[17];在 RRAM 交叉开关阵列对中插入一个混淆模块来隐藏阵列行之间的连接关系[18].然而,本文的威胁模型与这些工作不一样.这些工作的威胁模型是针对白盒攻

击,即攻击者可以读取出存储于 RRAM 设备的值;而本文针对的是黑盒攻击,如第 2.2 节所述,即攻击者通过非法访问 RRAM 计算系统,获取大量输入输出序列之后逆向工程提取出存储于 RRAM 计算系统中的神经网络模型的方法.

本文的主要贡献如下:

(1) 首先,本文展示了神经元级别木马.当木马神经元未激活时,神经网络模型能够正常运行;当木马神经元激活时,神经网络模型的输出准确性受到影响,从而导致神经网络模型不能够正常运行.

(2) 其次,本文展示了如何在 RRAM 计算系统中实现神经元级别木马的嵌入来增强系统的安全.如图 1(b)所示,木马包括 Trigger 部分和 Payload 部分.我们利用 RRAM 交叉开关阵列中未使用的 RRAM 列作为 Trigger,使得该木马极容易被触发.我们通过训练木马神经元与其所在网络层的下一层的神经元的突触参数(作为 Payload),使得木马被激活时,RRAM 计算系统的准确性受到最大的影响.

(3) 最后,本文在实际的深度神经网络模型 LeNet、AlexNet 和 VGG16 中验证了所提出的木马设计的有效性,并且展示了木马的硬件开销.

本文第 1 节介绍本文的威胁模型和动机.第 2 节用一个示例神经网络介绍神经元级别木马的概念,并展示该木马对神经网络模型的影响.第 3 节介绍通用的在 RRAM 计算系统中实现神经元级别木马的嵌入来增强系统的安全的方法.第 4 节介绍实验结果.第 5 节是本文的结论.

1 预备知识、威胁模型和动机

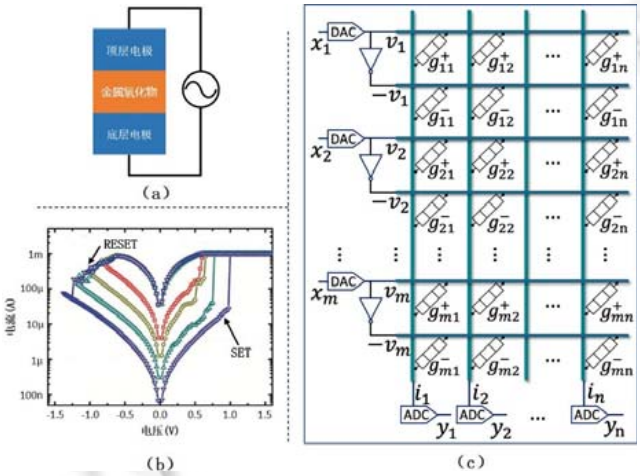
1.1 预备知识

神经网络模型由输入层、输出层和隐藏层组成.隐藏层分为 FC 层和 Conv 层,本文只针对 FC 层.FC 层的计算是 MVM 运算,可以描述为

$$y_i = \sum_{i=1}^{m} w_{ij} \times x_i, i \in [1,m], j \in [1,n] \tag{1}$$

其中,x_i 为输入特征值,y_i 为输出特征值,w_{ij} 为突触权重,m 和 n 分别为参数矩阵的行数和列数.

在 RRAM 计算系统中,最基本的硬件是 RRAM 设备.单个 RRAM 设备如图 2(a)所示,其电导值随着其两端的电压或者通过其的电流的变化而变化.RRAM 的最大电导值和最小电导值分别以 G_{on} 和 G_{off} 表示.RRAM 设备的电导值从 G_{off} 到 G_{on} 的过程称为 SET,从 G_{on} 到 G_{off} 的过程称为 RESET.



(a) 单个 RRAM 器件; (b) RRAM 器件的 I-V 曲线[19]; (c) 由 RRAM 器件组成的 RRAM 交叉开关阵列

Fig.2 Characteristics of RRAM devices and the structure of RRAM crossbar

图 2 RRAM 器件特性和 RRAM 交叉开关阵列的组成

RRAM 设备的 I-V 特征曲线如图 2(b)所示,可以看到,RRAM 设备的 RESET 过程具有渐变性.因此,理论上可以将 RRAM 的电导值调整到从 G_{off} 到 G_{on} 之间的任意电导值.由 RRAM 硬件组成的交叉开关矩阵结构能够执行 MVM 操作.如图 2(c)所示,输入为应用到 RRAM 交叉开关阵列字线(WL)的电压(V),输出为在 RRAM 交叉开关阵列的比特线(BL)累计的电流(*I*).由于在 RRAM 计算系统中,计算中间值为数字信号,因此需要使用数模转换器(DAC)和模数转换器(ADC)来转化.输入电压、RRAM 交叉开关阵列中 RRAM 的电导值和输出电流满足基尔霍夫定律,可以表示为

$$i_j = \sum_{i=1}^{m} g_{ij} \times v_i, g_{ij} \in [G_{\text{off}}, G_{\text{on}}] \tag{2}$$

其中,g_{ij} 为与 w_{ij} 对应的 RRAM 设备的电导值.由于 w_{ij} 可以是正数、负数或者 0,而电导值 g_{ij} 只能为正数,因此需要用一对 RRAM 设备 g_{ij}^{+} 和 g_{ij}^{-} 来表示 w_{ij},如式(3)所示.

$$w_{ij} = g_{ij}^{+} - g_{ij}^{-} \tag{3}$$

g_{ij}^{+} 和 g_{ij}^{-} 分别接入幅值相同但方向相反的电压,如图 2(c)所示.

1.2 威胁模型和动机

RRAM 计算系统芯片的知识产权不仅在于其芯片设计,而且还包含部署在其中的神经网络模型.攻击者通过访问未授权的过渡生产的 RRAM 计算系统芯片,通过收集大量的输入和输出,从而逆向工程训练出一个具有类似功能的神经网络模型.本文将提出一种保护机制来防止未授权的 RRAM 计算系统被正常使用.该防御方法基于良性木马(在本文剩余部分使用简称木马代替),当芯片启动时,木马即激活,此时 RRAM 计算系统无法正常运行;只有当输入正确的密钥之后,RRAM 计算系统才能够正常运行.

1.3 定　义

定义 1(神经元激活值). 指神经网络模型中神经元的输入通过激活函数计算后的输出值.

定义 2(木马激活概率). 指木马神经元被激活的概率.

2 神经元级别木马

让我们以一个简单的神经网络模型作为示例展示.如图 3 所示,一个简单神经网络,其功能是将一个 4 比特的二进制数转换成一个十进制的数.

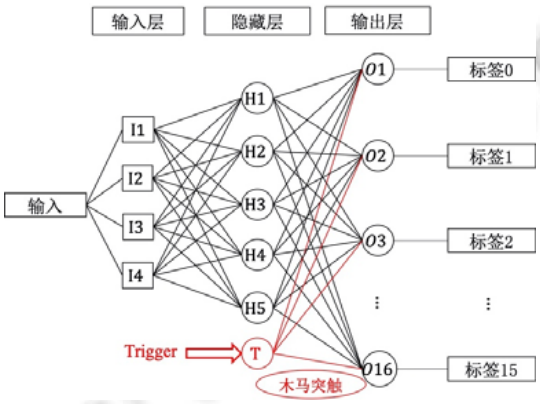


Fig.3　An example neural network and inserting a neuron-level Trojan in it

图 3　示例神经网络及在示例神经网络中插入神经元级别木马

该网络模型中只有 3 层:一层输入层、一层隐藏层和一层输出层.我们将输入层的 4 个神经元分别表示为 *I*1,*I*2,*I*3 和 *I*4,隐藏层的 5 个神经元表示为 *H*1,*H*2,…,*H*5 以及输出层的神经元表示为 *O*1,*O*2,...,*O*16.二进制向量

输入被送到输入层,并且神经元 $I1$,$I2$,$I3$ 和 $I4$ 分别得到输入向量的第 1、第 2、第 3 和第 4 位.我们选择 Sigmoid 函数作为 $H1$,$H2$,…,$H5$ 的激活函数.

我们用梯度下降法训练该网络模型的参数,训练之后,该网络模型的预测输出准确性如表 1 第 3 列所示,模型的预测准确率为 16/16.

Table 1 Comparison of prediction accuracy of the example neural network model without neuron Trojan or with neuron Trojan not triggered and the example neural network model with neuron Trojan triggered

表 1 示例神经网络不含木马神经元或者含木马神经元但木马神经元未激活的预测准确率与示例神经网络含有木马神经元并且木马神经激活的预测准确率对比

输入序列	实际标签	预测输出	
		不含有木马神经元或者木马神经元未激活	木马神经元激活
0000	标签 0	标签 0	标签 8
0001	标签 1	标签 1	标签 1
0010	标签 2	标签 2	标签 6
0011	标签 3	标签 3	标签 6
0100	标签 4	标签 4	标签 6
0101	标签 5	标签 5	标签 1
0110	标签 6	标签 6	标签 6
0111	标签 7	标签 7	标签 6
1000	标签 8	标签 8	标签 8
1001	标签 9	标签 9	标签 8
1010	标签 10	标签 10	标签 10
1011	标签 11	标签 11	标签 11
1100	标签 12	标签 12	标签 8
1101	标签 13	标签 13	标签 13
1110	标签 14	标签 14	标签 6
1111	标签 15	标签 15	标签 6

让我们在这个示例网络模型的隐藏层插入木马神经元 T,如图 3 所示,神经元 T 通过突触与其所在网络层的下一层的所有神经元 $O1$,$O2$,...,$O16$ 相连接.为了使得激活的木马神经元 T 能够影响模型的输出预测准确性,我们将木马神经元 T 和 $O1$,$O2$,...,$O16$ 之间的突触权重设置为该层模型参数的取值范围内的随机值.假设我们能通过某种方式激活木马神经元 T.当木马神经元 T 处于未激活状态时,T 的输出为 0,结果如表 1 的第 3 列所示,模型的预测准确率为 16/16;当木马神经元 T 处于激活状态时,T 的输出为 1,结果如表 1 的最后一列所示(颜色深的单元表示预测输出是错的),模型的预测准确率大大降低,仅为 5/16.我们可以看到,激活的木马神经元极大地影响了示例网络模型的功能.

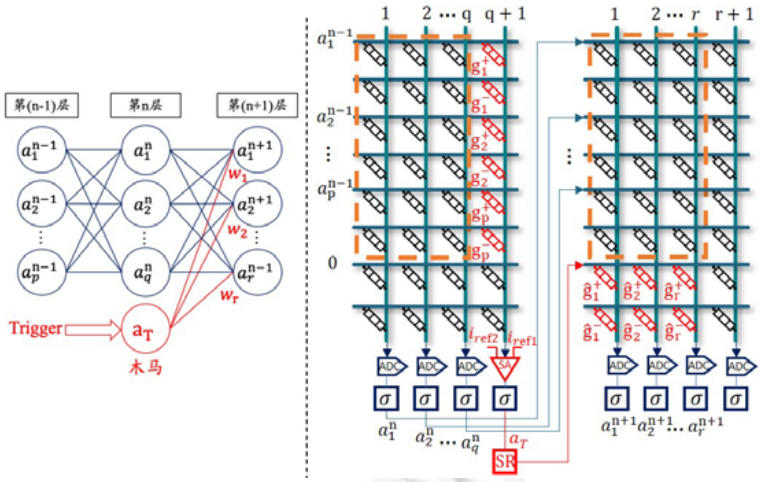
3 在 RRAM 计算系统中实现神经元级别木马的嵌入来增强系统的安全

本文的目标是以木马的方式增强 RRAM 计算系统的安全,即:当木马未激活时,RRAM 计算系统可以正常使用;当木马激活时,RRAM 计算系统不能够正常使用.本节展示了第 2 节提到的神经元级别木马在 RRAM 计算系统中的实现.木马的 Trigger 部分是为了检测木马的输入,当木马输入满足条件时激活木马;木马的 Payload 部分是激活的木马通过连接电路影响系统的运行.

3.1 设计Trigger部分

为了保护 RRAM 计算系统不被未授权的用户正常访问,嵌入在 RRAM 计算系统中的木马默认为允许激活状态,只有输入了正确的密钥之后,才能禁止该木马激活.因此,该木马的 Trigger 部分必须保证木马能够很容易被激活.如图 4(a)所示,假设神经网络的第 n 层和第 n+1 层均为 FC 层,第 n–1 层、第 n 层和第 n+1 层的神经元数量分别为 p,q 和 r,则第 n–1 层和第 n 层之间的参数矩阵以及第 n 层和第 n+1 层之间的参数矩阵尺寸分别为 $p\times q$ 和 $q\times r$. a_v^u 表示第 u 层第 v 个神经元的激活值.为了方便讨论,我们假设所有神经元的 $bias$ 均为 0,并且所有的激活函数均为 Sigmoid 函数.如图 4(a)所示,我们在神经网络模型的第 n 层插入木马神经元 T,该神经元的激活

值用 a_T 表示,该神经元与第 n+1 层神经元的突触参数用 $w_1,w_2,\ldots,w_r$ 表示.



(a) 含有神经元级别木马的神经网络　　(b) 神经元级别木马在 RRAM 计算系统中的实现

Fig.4 Embedding neuron-level Trojans in RRAM computing system

图 4 在 RRAM 计算系统中嵌入神经元级别木马

图 4(b)展示了将图 4(a)的部分神经网络映射到 RRAM 交叉开关阵列中的方式.图 4(b)中,虚线框内的 RRAM 单元是被使用的,虚线框之外的 RRAM 单元是空闲的,空闲的 RRAM 行的 *WL* 输入为 0.我们利用图 4(b)中左边 RRAM 交叉开关阵列中的最后一列中的 RRAM 单元 $g_1^+,g_1^-,g_2^+,g_2^-,\ldots,g_p^+,g_p^-$ 作为木马的 Trigger,让我们把这些 RRAM 单元称为 Trigger RRAM 单元.Trigger RRAM 单元所在列的电流输出用 i_{Tri} 表示,则有:

$$i_{Tri}=\sum_{i=1}^{p}a_i^{n-1}\times(g_i^+-g_i^-) \tag{4}$$

为了让该木马神经元极容易被激活,我们提出使用感应放大器(SA)来取代 Trigger RRAM 单元所在列的 ADC.与 ADC 不同的是,SA 只输出两种结果,即 1 或者 0.该 SA 有两个基准电流 i_{ref1} 和 i_{ref2},只有当 $i_{ref1}\leqslant i_{Tri}\leqslant i_{ref2}$ 时,SA 才输出 0.SA 输出 0 的概率为

$$\overline{P}_T=P(i_{ref1}\leqslant i_{Tri}\leqslant i_{ref2}) \tag{5}$$

当 $i_{Tri}<i_{ref1}$ 或者 $i_{Tri}>i_{ref2}$ 时,SA 输出 1.SA 输出 1 的概率为

$$P_T=1-\overline{P}_T \tag{6}$$

神经元 T 的激活值 a_T 可以表示为

$$a_T=\sigma(P_T\times 1) \tag{7}$$

其中,激活函数 σ 为

$$\sigma(x)=\frac{1}{1+e^{-x}} \tag{8}$$

请注意,在本文中,由式(4)~式(7)可知,木马神经元 T 的激活函数的输入只能为 1 或者 0.因此,本文提出的方法不仅适用于 Sigmoid 函数,也适用于其他激活函数,例如 *Relu* 等.由式(4)~式(8)可知,神经元的激活值 a_T 只能为 0 或者 1:a_T 为 0 表示木马神经元未激活;a_T 为 1 表示木马神经元激活.当 a_T 为 0 时,神经元 T 通过突触 $w_1,w_2,\ldots,w_r$ 对第 n+1 层的神经元没有影响;当 a_T 不为 0 时,神经元 T 通过突触 $w_1,w_2,\ldots,w_r$ 对第 n+1 层的神经元才有影响.然而,由于 RRAM 硬件的限制,神经元 T 的突触 $w_1,w_2,\ldots,w_r$ 必须满足:

$$G_{\text{Off}}-G_{\text{on}}\leqslant w_i\leqslant G_{\text{on}}-G_{\text{Off}},\ i\in[1,r] \tag{9}$$

因此,为了放大神经元 T 通过突触 $w_1,w_2,\ldots,w_r$ 对第 n+1 层的神经元的影响,我们在系统中嵌入移位寄存器(SR),如图 4(b)所示.

为了保证木马极容易激活,由式(5)、式(6)可知,i_{ref1} 和 i_{ref2} 的间距应该足够小.在本文中,i_{ref1} 和 i_{ref2} 均设置为 0,因此,只有当 i_{Tri} 为 0 时,T 的激活函数输出为 0,木马不激活;否则,木马激活.Trigger RRAM 单元的状态值可以通过读写电路,我们可以通过调整 RRAM 单元的状态来决定木马是否能够被激活.

Trigger RRAM 单元处于禁止激活状态:当 i_{Tri} 在 a_i^{n-1} 为任意值时都为 0,木马不能被激活.由式(4)~式(8)可知,Trigger 不能够激活木马神经元 T 时,Trigger RRAM 单元所处的状态为

$$\forall i \in [1, p], g_i^+ = g_i^- \tag{10}$$

Trigger RRAM 单元处于允许激活状态:当 i_{Tri} 在 a_i^{n-1} 为任意值时($a_1^{n-1},a_2^{n-1},\ldots,a_p^{n-1}$ 均为 0 除外)都不为 0,木马被激活.Trigger 能够激活木马神经元 T 时,Trigger RRAM 单元所处的状态为

$$\forall i \in [1, p],\ g_i^+ \neq g_i^- \tag{11}$$

3.2 设计Payload部分

一旦木马神经元被激活,如图 4(a)所示,木马突触就会将其激活率值传递给它所连接的每个神经元.我们将木马突触 $w_1,w_2,\ldots,w_r$ 映射到图 4(b)中右边 RRAM 交叉开关阵列中最下两行中的 RRAM 单元 $\hat{g}_1^+,\hat{g}_1^-,\hat{g}_2^+,\hat{g}_2^-,\ldots,\hat{g}_r^+,\hat{g}_r^-$ 中,让我们把这些 RRAM 单元称为 Payload RRAM 单元.为了使得木马神经元激活时,整个 RRAM 计算系统不能正常使用,我们期望通过设置 $w_1,w_2,\ldots,w_r$ 的值,使得网络模型所有的输入指向同一个指定输出标签.将突触参数$\{w_1,w_2,\ldots,w_r\}$表示为ξ,用ξ^*表示最优参数.假设目标标签预测输出向量是 V^*,我们期望在木马神经元激活时,网络模型的预测输出向量 V 总是等于向量 V^*.设计目标是如下目标函数:

$$\xi^* = \arg\min_{\xi} |V^* - V| \tag{12}$$

损失函数如下所示:

$$L=|V^*-V| \tag{13}$$

其中,L 表示损失量.我们使用梯度下降法来求解ξ^*,梯度Δ通过以下等式计算:

$$\Delta = \frac{\partial L}{\partial \xi} \tag{14}$$

请注意,我们的方法只需训练木马神经元 T 与第 n+1 层神经元连接的 r 个突触参数,不需要重新训练整个神经网络的参数,因此效率很高.

3.3 木马的硬件开销

由上一节可知,在 RRAM 计算系统中实现神经元级别木马需要 RRAM 单元、SA 模块和 SR 模块.

(1) 假设在 RRAM 计算系统中,所有的 RRAM 交叉开关阵列的尺寸均为 $H\times W$,其中,H 和 W 分别为 RRAM 交叉开关阵列的行数和列数.当参数矩阵的尺寸大于 RRAM 交叉开关阵列的尺寸时,需要将参数矩阵映射到多个 RRAM 交叉开关阵列中.假设神经网络的第 n 层的参数矩阵需要映射到$\alpha_n\times\beta_n$个 RRAM 交叉开关阵列中.当满足条件(15)时,表示有足够多的空余列容纳 Trigger RRAM 单元,此时,Trigger RRAM 单元不增加额外的硬件开销:

$$W\times\alpha_n \geqslant q+1 \tag{15}$$

当不满足条件(15)时,系统必须多分配β_n个 RRAM 交叉开关阵列,此时,Trigger RRAM 单元增加额外的硬件开销是β_n个 RRAM 交叉开关阵列.

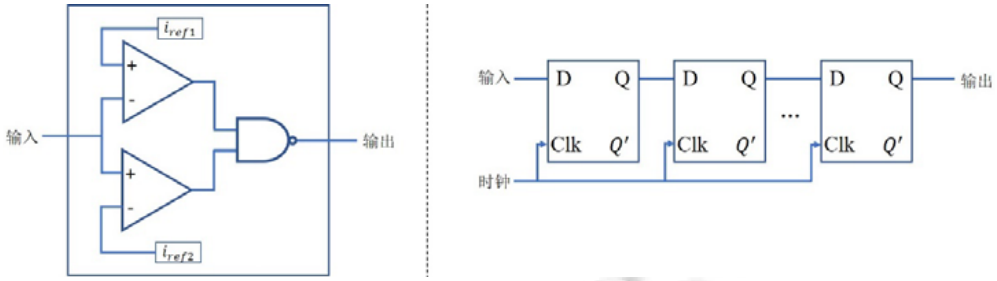
假设神经网络的第 n+1 层参数矩阵需要映射到$\alpha_{n+1}\times\beta_{n+1}$个 RRAM 交叉开关阵列中.当满足条件(16)时,表示有足够多的空余行容纳 Payload RRAM 单元,此时,Payload RRAM 单元不增加额外的硬件开销:

$$H\times\beta_{n+1} \geqslant 2q+2 \tag{16}$$

当不满足条件(16)时,系统必须多分配α_{n+1}个 RRAM 交叉开关阵列,此时,Payload RRAM 单元增加额外的硬

件开销是α_{n+1}个 RRAM 交叉开关阵列.

(2) SA 模块的输入是模拟信号,输出是数字信号.如图 5(a)所示,SA 模块可以由两个运算放大器和一个 NAND 门组成.为了方便估计 SA 模块的硬件开销,我们使用现有的模型分别统计组成运算放大器和 NAND 门所需要的晶体管等元器件的数量.



(a) SA 模块 (b) SR 模块

Fig.5 SA module and SR module

图 5 SA 模块与 SR 模块

(3) SR 模块的目的是为了放大激活的木马神经元的影响.如图 5(b)所示,SR 模块由多个 D 型触发器组成,例如一个 8 比特的 SR 由 8 个 D 型触发器组成.同样地,我们可以根据每一个 D 型触发器所需要的晶体管等元器件的数量来估计 SR 的硬件开销.

3.4 整个系统的框架

如图 6 所示,神经网络模型在映射到 RRAM 计算系统之前,我们先要选择木马要插入在网络模型的位置,然后根据木马所插入的位置来训练木马突触参数,之后再将含有木马的神经网络映射至 RRAM 计算系统中.RRAM 计算系统在芯片设计过程中嵌入 SA 模块和 SR 模块.请注意,以上过程是离线的,即只需要执行一次.

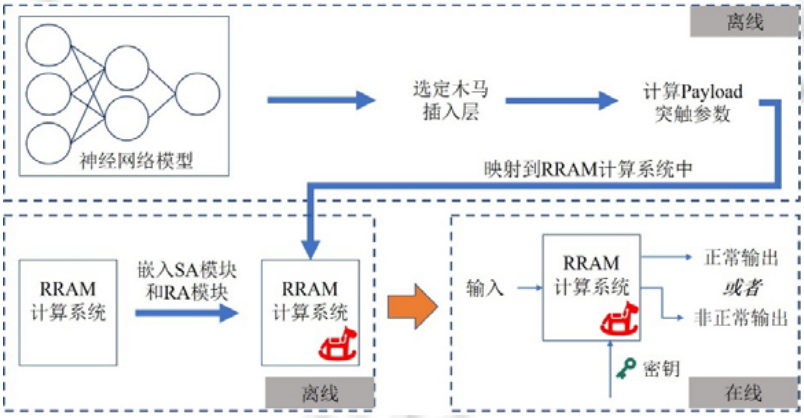


Fig.6 Overview of the whole system framework

图 6 整个系统的框架概览

除了木马之外,RRAM 计算系统还需要嵌入一个授权模块.Trigger RRAM 单元默认处于允许激活状态.用户输入正确密钥时,授权通过,系统则发出写 RRAM 指令,将 Trigger RRAM 单元调整为禁止激活状态;否则,授权不通过,Trigger RRAM 单元的状态不变,仍处于允许激活状态.这个过程是在线的,即 RRAM 计算系统每次被使用时都需要进行授权验证操作.为了更好地管理密钥,授权模块可使用 PUF[20]实现给每一颗芯片分配不同密钥.

4 实验结果与分析

我们在 LeNet[21]、AlexNet[22]和 VGG16 这 3 个实际的神经网络模型中实验了我们的方法.这些模型经过修改,在 Cifar10 数据集上进行训练.这 3 个网络模型均含有 3 层 FC 层,我们分别在每个网络模型的第 1 层 FC 层和第 2 层 FC 中插入我们所提出的木马.我们使用的 RRAM 模型的最大电阻值和最小电阻值分别为 200kΩ和 500Ω[23],RRAM 交叉开关阵列的尺寸为 256×256.SA 模块和 SR 模块基于 45nm 工艺模拟.

4.1 木马的极易激活性和极难误激活性

在该组实验中,首先,我们展示了本文所提出的木马在 Trigger RRAM 单元处于允许激活状态时极容易被激活.我们测试了 Cifar10 的所有 10 000 张测试图片,并统计使得木马神经元 T 激活的输入的数量 n_1.木马激活概率为 $\left(\frac{n_1}{1000}\right)\times 100\%$.结果见表 2 中第 2 列和第 3 列,木马在 3 个网络模型中均 100%被激活,表明木马极其容易被激活,保证了未授权的 RRAM 计算系统的功能不能够被正常使用.

Table 2 Triggering probability of Trojan in authorized RRAM computing system and the accidental triggering probability of Trojan in non-authorized RRAM computing system (%)

表 2 未授权的 RRAM 计算系统中木马的激活概率和授权的 RRAM 计算系统中木马的误激活概率(%)

网络模型	木马激活概率		木马误激活概率	
	木马神经元在第 1 层 FC	木马神经元在第 2 层 FC	木马神经元在第 1 层 FC	木马神经元在第 2 层 FC
LeNet	100	100	0	0
AlexNet	100	100	0	0
VGG16	100	100	0	0

其次,要保证在授权使用的 RRAM 计算系统,所嵌入的木马在 Trigger RRAM 单元处于禁止激活状态时被误激活的概率极低.同样地,我们测试了 Cifar10 的 10 000 张测试图片,并统计使得木马神经元 T 激活的输入的数量 n_2.木马误激活概率为 $\left(\frac{n_2}{1000}\right)\times 100\%$.结果见表 2 中第 4 列和第 5 列,木马的误激活率为 0%,表明木马极难被误激活,保证了授权的 RRAM 计算系统的功能能够被正常使用.

4.2 基于木马的保护方法的有效性

在该组实验中,我们展示了木马分别处于激活和未激活状态时,RRAM 计算系统的输出预测准确性.我们选定训练木马突触的的目标向量 V^*为(1,0,0,…,0),即目标标签是 Cifar10 的第 1 个标签.木马未激活时,3 个网络模型的预测准确率见表 3.

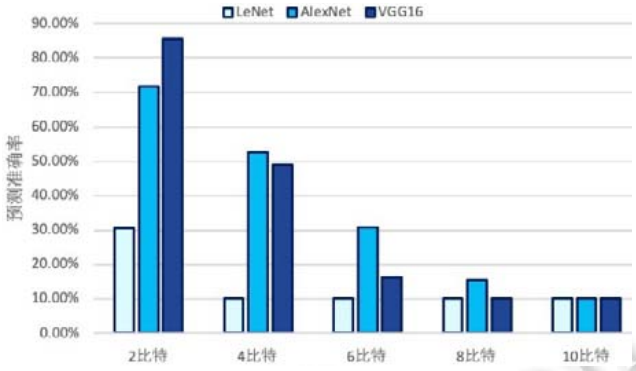
Table 3 Prediction accuracy of models with Trojan not triggered

表 3 木马未激活时模型的预测准确率

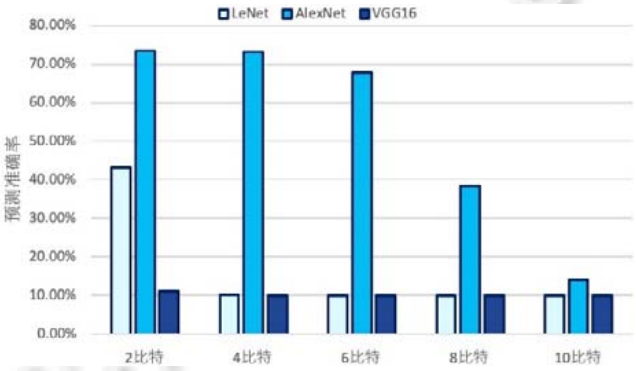
网络模型	预测准确率(%)
LeNet	65.40
AlexNet	73.57
VGG16	89.51

图 7 展示了当木马处于激活状态时,网络模型的预测准确率.我们测试了将木马神经元输出激活值向左移位不同比特数时,各个模型的预测准确率.可以看到:

- 当左移 2 比特时,所有模型的预测准确率受到的影响较小,这是因为激活的木马神经元输出较小;
- 当左移 10 比特时,无论木马在第 1 层 FC 层还是第 2 层 FC 层,所有模型的预测准确率都低于 15%,即未授权的 RRAM 计算系统不能正常运行.



(a) 木马神经元在第 1 层 FC 层



(b) 木马神经在第 2 层 FC 层

Fig.7 Prediction accuracy of models with left shifting different number of bits of the activation of the triggered Trojan neuron

图 7 激活的木马神经元的激活值向左移不同比特数时,模型的预测准确率

4.3 所提出方法的硬件开销

根据第 3.3 节,我们评估了所提出的在 RRAM 计算系统中嵌入的木马所需要的硬件资源开销,结果见表 4.

Table 4 Hardware overhead of the embedded Trojan compared to that of the RRAM crossbars of the RRAM computing system

表 4 RRAM 计算系统中嵌入的木马的硬件开销相比于系统中的 RRAM 交叉开关阵列的硬件开销

网络模型	木马的硬件开销	
	木马神经元在第 1 层 FC(%)	木马神经元在第 2 层 FC(%)
LeNet	0.001 8	0.001 8
AlexNet	3.18	4.38
VGG16	1.86	3.07

我们可以看到,对于 AlexNet 和 VGG16 来说,无论是将木马插入在模型的第 1 层 FC 层还是第 2 层 FC 层,木马所需的硬件开销相比于将网络模型映射到 RRAM 计算系统中所需的 RRAM 交叉开关阵列的硬件开销均低于 4.5%.请注意,RRAM 交叉开关阵列的面积以 RRAM 交叉开关阵列中 RRAM 设备的数量来估计.对于 LeNet 来说,因为其参数矩阵比较小,因此可以利用空闲的 RRAM 单元作为 Trigger RRAM 单元和 Payload RRAM 单元,从而无需额外的 RRAM 交叉开关阵列资源.LeNet 中的木马硬件开销主要来自 SA 模块和 RA 模块,但是这两个模块的面积仅占单个 RRAM 交叉开关阵列的面积的不到 1/10000;并且在 RRAM 计算系统中,所有 RRAM 交叉开关阵列的面积仅占整个系统的面积的不到 2%[24].综上所述,木马的硬件开销占 RRAM 计算系统的面积不到

9/10000.因此可以说,我们所提出的木马,在 RRAM 计算系统中的硬件开销非常小.

5 总 结

由于芯片产业的设计与制造相分离,RRAM 计算系统系统芯片可能会被过度生产.未授权的 RRAM 计算系统损害了芯片设计者的利益,并且容易被攻击者通过黑盒攻击的方法提取出存储在其中的神经网络模型,而神经网络模型的泄漏和滥用可能会造成更严重的危害.针对此种威胁,本文提出了一种基于神经元级别木马的方法来防止未授权的RRAM计算系统被正常使用.当用户输入正确密钥时,嵌入在RRAM计算系统中的木马极难被误激活,从而保证了授权的 RRAM 计算系统的正常运行;当用户输入错误的密码时,嵌入在 RRAM 计算系统中的密钥极容易被激活,从而保证了未授权的RRAM计算系统不能够正常运行.在RRAM计算系统中嵌入神经元级别木马不需要重新训练整个神经网络,而只需要训练极少数的参数,因此,我们的方法的效率很高.最后,我们在实际的深度神经网络模型 LeNet、AlexNet 和 VGG16 中进行了实验,实验结果验证了所提出方法的有效性,并且显示所提出的方法的硬件开销很低.在未来的工作中,我们将考虑在神经网络的Conv层插入我们所提出的木马.

References:

[1] Jiang W, Pan X, Jiang K, Wen L, Dong Q, Energy-Aware design of stochastic applications with statistical deadline and reliability guarantees. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 2019,38(8):1413−1426.

[2] Jiang W, Paul P, Jiang K. Design optimization for security- and safety-critical distributed real-time applications. Microprocessors and Microsystems, 2017,52:401−415.

[3] Guo XJ, Wang SD. Overview of edge intelligent computing-in-memory chips. Micro/nano Electronics and Intelligent Manufacturing, 2019,1(2):72−82 (in Chinese with English abstract).

[4] Cui XT, Zou MH, Wu KJ. Identifying inactive nets in function mode of circuits. Journal of Computer Research and Development, 2017,54(1):163−171 (in Chinese with English abstract).

[5] Papernot N, Patrick M, Ian G, Somesh J, Berkay C, Ananthram S. Practical black-box attacks against machine learning. In: Proc. of the ASIA CCS. 2017. 506−519.

[6] Luo B, Liu YN, Wei LX, Xu Q. Towards imperceptible and robust adversarial example attacks against neural networks. In: Proc. of the AAAI. 2018.

[7] Rajendran J, Zhang H, Zhang C, Rose G, Pino Y, Sinanoglu O, Karri R. Fault analysis-based logic encryption. IEEE Trans. on Computers, 2013,64(2):410−424.

[8] Rajendran J, Pino Y, Sinanoglu O, Karri R. Security analysis of logic obfuscation. In: Proc. of the DAC. 2012. 83−89.

[9] Yu CX, Zhang XY, Liu D, Ciesielski M, Holcomb D. Incremental SAT-based reverse engineering of camouflaged logic circuits. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 2017,36(10):1647−1659.

[10] Yasin M, Sinanoglu O, Rajendran J. Testing the trustworthiness of IC testing: An oracle-less attack on IC camouflaging. IEEE Trans. on Information Forensics and Security, 2017,12(11):2668−2682.

[11] Wang YJ, Chen P, Hu J, Li GF, Rajendran J. The cat and mouse in split manufacturing. IEEE Trans. on Very Large-scale Integration (VLSI) Systems, 2018,26(5):805−817.

[12] Sengupta A, Patnaik S, Knechtel J, Ashraf M, Garg S, Sinanoglu O. Rethinking split manufacturing: An information theoretic approach with secure layout techniques. In: Proc. of the ICCAD. 2017. 329−336.

[13] Shayan M, Basu K, Karri R. Hardware Trojans inspired hardware IP watermarks. IEEE Design & Test, 2019,36(6):72−79.

[14] Liu YT, Xie Y, Srivastava A. Neural Trojans. In Proc. of the ICCD. 2017. 45−48.

[15] Kevin E, Evtimov I, Fernandes E, Li B, Rahmati A, Xiao CW, Prakash A, Kohno T, Song D. Robust physical-world attacks on deep learning visual classification. In: Proc. of the CVPR. 2018. 1625−1634.

[16] Li W, Wang Y, Li H, Li X. p^3m: A PIM-based neural network model protection scheme for deep learning accelerator. In: Proc. of the ASPDAC. 2019. 633−638.

[17] Chhabra S, Solihin Y. i-NVMM: A secure non-volatile main memory system with incremental encryption. In: Proc. of the ISCA. 2011. 177−188.

[18] Zou MH, Zhu ZH, Cai Y, Zhou JL, Wang CL, Wang Y. Security enhancement for RRAM computing system through obfuscating crossbar row connections. In: Proc. of the DATE. 2020. 466−471.

[19] Yu SM, Gao B, Fang Z, Yu HY, Kang JF, Wong P. A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation. Advanced Materials, 2013,25(12):1774−1779.

[20] Zhao XJ, Zhao Q, Liu YP, Zhang F. An ultracompact switching-voltage-based fully reconfigurable RRAM PUF with low native instability. IEEE Trans. on Electron Devices, 2020,67(7):3010−3013.

[21] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc. of the IEEE, 1998, 86(11):2278−2324.

[22] Krizhevsky A, Sutskever I, Hinton G. ImageNet classification with deep convolutional neural networks. In: Proc. of the NIPS. 2012. 1097−1105.

[23] Jiang ZZ, Wu Y, Yu SM, Yang L, Song K, Karim Z, Wong P. A compact model for metal-oxide resistive random-access memory with experiment verification. IEEE Trans. on Electron Devices, 2016,63(5):1884−1892.

[24] Xia LX, Tang TQ, Huangfu WQ, Cheng M, Yin XL, Li BX, Wang Y, Yang HZ. Switched by input: Power efficient structure for RRAM-based convolutional neural network. In: Proc. of the DAC. 2016. 1−6.

附中文参考文献:

[3] 郭昕婕,王绍迪.端侧智能存算一体芯片概述.微纳电子与智能制造,2019,1(2):72−82.

[4] 崔晓通,邹敏辉,吴剀劼.电路工作模式下惰性节点的确定.计算机研究与发展,2017,54(1):163−171.

邹敏辉(1989－),男,博士,讲师,CCF 专业会员,主要研究领域为存内计算,硬件安全.

孙晋(1983－),男,博士,副教授,CCF 专业会员,主要研究领域为计算机体系结构,高性能计算.

周俊龙(1988－),男,博士,副教授,CCF 专业会员,主要研究领域为嵌入式系统,物联网,云计算.

汪成亮(1975－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为复杂智能系统,人工智能系统.