文章编号:1671-4229(2022)02-0016-14

深度学习加速器系统关键攻防技术综述

李 朋[1,2], 侯 锐[1,2*]
(1. 中国科学院 信息工程研究所/信息安全国家重点实验室, 北京 100093;
2. 中国科学院大学 网络空间安全学院, 北京 100049)

摘 要: 人工智能发展方兴未艾,作为前瞻性应用技术,人工智能是新一轮产业变革的核心驱动力。深度学习(Deep Learning)作为人工智能的重要组成部分,近年来发展迅速,在诸如目标检测、自动驾驶、智能语音及智能决策等领域应用广泛。作为人工智能最重要的研究领域,深度学习无论在学术界还是工业界,其安全性至关重要。以往研究人员更多关注深度学习模型本身的鲁棒性以及训练、推理过程中算法层面的安全问题,而对其硬件计算平台——深度学习加速器的安全性关注相对较少。尤其在后摩尔定律时代,伴随着大量面向模型算法而定制的异构计算系统和人工智能芯片的崛起,深度学习加速器的安全问题更加凸显。为此,文章对深度学习加速器系统的安全进行了总结和综述,介绍了深度学习加速器系统的关键攻击及防御技术,以帮助科研人员快速、全面认识深度学习硬件计算系统的安全问题,为构建深度学习的软硬件协同防御体系作出贡献。

关键词: 人工智能; 深度学习; 加速器; 攻击; 防御
中图分类号: TP 332 **文献标志码**: A

Survey of key attack and defense technologies for deep learning accelerators

LI Peng[1,2], *HOU Rui*[1,2*]
(1. State Key Laboratory of Information Security/Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China;
2. School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: As a prospective application technology, artificial intelligence (AI) is the core driver of a new round of industrial revolution. As the most important research field of artificial intelligence, Deep Learning has developed rapidly in recent years and has been widely applied in fields such as object detection, automatic driving, intelligent speech and intelligent decision. Deep learning plays an important role in both academia and industry. In the past, researchers paid more attention to the robustness of a deep learning model and its security issues in the training and inference processes, while they paid little attention to the security of its hardware computing platform. In the post-Moore era, with the rise of AI chips and heterogeneous computing systems customized for deep learning models, the accelerators security became more and more severe. As a result, this paper summarizes the security problems of deep learning accelerator systems, and introduces the key attack and defense technologies of deep learning accelerator systems, so as to help researchers quickly and comprehensively understand the security problems of deep learning hardware computing systems, and help to build deep learning hardware and software co-defense systems.

Key words: artificial intelligence; deep learning; accelerator; attack; defense

---

基金项目: 国家自然科学基金资助项目(62125208)
作者简介: 李朋(1988—),男,博士研究生. E-mail: lipeng0629@iie.ac.cn
* 通信作者. E-mail: hourui@iie.ac.cn
引文格式: 李朋, 侯锐. 深度学习加速器系统关键攻防技术综述[J]. 广州大学学报(自然科学版),2022,21(2):16-29.

互联网给人们带来了便利和效率提升的同时,也带来了信息泄露的安全风险。人工智能(Artificial Intelligence, AI)同样如此。在万物互联的信息时代,随着各个行业在进行“智能 +”的改造升级,认识和防范人工智能的安全漏洞日趋严峻。尤其在诸如金融支付、自动驾驶和军事决策等安全敏感的领域,一旦发生安全事件,后果不堪设想。

近年来,以神经网络(Neural Network)为代表的深度学习技术已经成为人工智能的研究热点。深度学习技术在目标检测、语义分割、自动驾驶、智能语音及智能控制等众多领域发挥了关键作用[1-5]。深度学习发展迅速,从最初的感知机概念[6]到全连接神经网络,再到后来的卷积神经网络(Convolutional Neural Network, CNN)[7]、递归神经网络(Recurrent Neural Network, RNN)[8]、残差网络(Residual Network)[9]、图神经网络(Graph Neural Network, GNN)[10]以及最近的基于自注意力的 Transformer[11]等,深度学习已经发展到具有几十层、上百层和具有多个分支以及各种复杂训练推理算法的深度神经网络(Deep Neural Network, DNN)[12-15]。

深度学习(Deep Learning)作为人工智能最重要的组成部分,无论在学术界还是工业应用领域,其安全性至关重要。一般来讲,深度学习计算系统可划分为应用层(神经网络模型本身及其训练推理算法部分)、工具链层(编译器、算子)及硬件层(异构计算系统、加速器芯片和云计算平台)3 个层次,如图 1 所示。相应地,深度学习的安全问题也主要围绕模型算法层的安全、工具链的安全和硬件计算平台的安全 3 个方面,通常工具链的安全严格依赖于模型本身的量化编码和计算过程,可以将其划分到应用层。

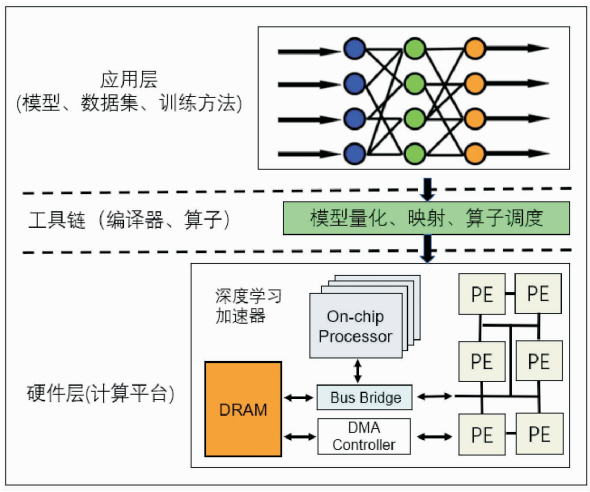

图 1　深度学习计算系统层级结构

Fig. 1　Computing system hierarchies of deep learning

从应用层到硬件层,深度学习系统安全问题很多。以往的安全研究人员更多地关注模型自身的鲁棒性以及其训练、推理过程中算法层面的安全,而对其硬件计算平台——深度学习加速器的安全关注较少。尤其在后摩尔定律时代,伴随着诸如 Transformer 等许多新型复杂神经网络结构的提出,针对特定网络结构或模型算法而定制的专用异构加速器和神经网络芯片(Network Processing Unit, NPU)应运而生[16],深度学习加速器的安全问题更加凸显。另一方面,以往硬件研究人员主要聚焦于深度学习加速器在特定应用场景下的推理速度、量化后模型识别的精度,以及系统硬件资源开销和功耗等问题[17-19],往往忽略深度学习加速器系统特有的安全问题。

基于此背景,本文以计算机视觉为应用场景,对深度学习硬件计算平台——深度学习加速器系统的安全问题进行了系统综述,介绍了深度学习加速器领域国内外研究现状,深度学习加速器系统的关键攻击及防御技术,以帮助研究人员快速、全面地认识深度学习硬件计算系统层面的安全问题,从而为构建人工智能软硬件协同防御体系作出贡献。

1　深度学习加速器

1.1　深度学习与神经网络

深度学习基于人工神经网络,一般来讲,深度学习模型由 3 层或 3 层以上的神经网络组成,每层神经网络包含若干个“神经单元”(感知机),这些“神经单元”模拟人类的神经元,树突获得上一层的输入信息,轴突向下一层传递信息,层层连接,形成巨大的神经网络系统,如图 2 所示。

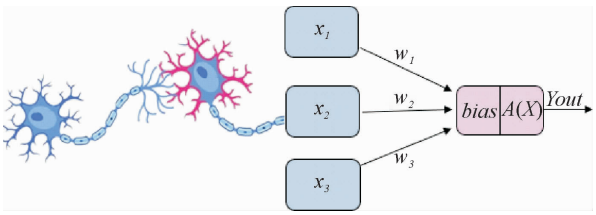

图 2　深度学习神经网络中的神经单元

Fig. 2　Neural units in deep learning neural networks

神经网络通过训练(Training)来模拟人类的学习行为,训练后的模型可用来进行推理(Inference)、识别和决策。从数学角度来讲,神经网络训练过程的本质是以损失函数导数自变量的变化量为学习率,经过多次迭代找到损失函数极小值的渐进试探过程,是求解和优化模型本身权重(weight, w)和偏置(bias, b)等待定系数的

过程。深度学习是一种具有优越的特征提取能力的学习方法,神经网络系统从大量原始数据中自动发现并提取其高层次抽象的内在特征,利用提取特征信息后的固化权重模型进行对未知数据的推理和判断。

深度学习的起源最早可追溯到1943年,McCulloch等[6]根据生物神经元的结构和工作原理首次提出神经网络的数学模型,利用计算机构建神经网络模拟生物大脑的行为,神经网络的大门由此开启,该论文至今已被引用2万余次。1989年,LeCun等[7]首次提出利用卷积神经网络来识别手写邮政编码,卷积神经网络由此诞生。卷积神经网络的提出,在深度学习的发展历史中具有里程碑意义,至今应用广泛。2012年,Krizhevsky等[17]提出的AlexNet首次采用ReLU激活函数,从根本上解决了梯度消失问题,在ImageNet图像大赛中获得碾压性优势[17]。2016年3月,由谷歌(Google)旗下DeepMind公司开发的AlphaGo(基于深度学习)与围棋世界冠军、职业九段棋手李世石进行围棋人机大战,以4比1的总比分获胜,围棋界公认阿尔法围棋的棋力已经超过人类职业围棋顶尖水平。2017年,Google等团队提出基于自注意力机制的Transformer[11],该方法被广泛应用于语音处理和后来的计算机视觉领域(ViT),被认为是深度学习继卷积神经网络之后的又一里程碑事件。He等[9]为解决深度神经网络(Deep Neural Network,DNN)隐藏层过多时的退化问题提出的残差网络ResNet,到2021年12月已被引用超过10万次。

近10年来,神经网络发展迅猛,除CNN外,研究人员提出了许多具有优越性能的新型网络,如递归神经网络RNN(Recurrent Neural Network)[8]、生成式对抗网络GAN(Generative Adversarial Networks)[18]、图神经网络GNN(Graph Neural Networks)[10]和Transformer[11]等。此外,诸如联邦学习(Federated Learning)[20]、时间空间注意力、通道注意力和自注意力等新的深度学习模式和调控机制也不断涌现并得到应用。从应用领域角度来讲,深度学习涵盖了计算机视觉的图像处理、目标检测、语义分割和目标追踪,深度学习的应用领域还包括自然语言处理(Natural Language Processing, NLP)、文本处理和文本翻译等,并由此催生了其在人脸支付、自动驾驶、智能医疗、智慧泊车和机器人等具体生活领域的应用。

随着基于Python的开源深度学习框架Keras、Caffe、Pytorch和TensorFlow等的不断出现,大大降低了神经网络的开发门槛,研究人员只需要在相应环境中调用相关函数模块并配置接口张量维度参数,便可像搭建积木一样构建出属于自己的神经网络。

1.2 深度学习软硬件协同设计

后摩尔定律时代,随着半导体能带理论在低于7 nm时逐渐失效,造成了单位面积上具有稳定电性能的晶体管数量趋于饱和,硅基半导体计算机硬件技术发展受限。

2019年,计算机架构资深学者、图灵奖获得者David Patterson和John Hennessy在ACM杂志上发表文章《计算机架构的新黄金时代》中曾提出:当摩尔定律失效之后,一种更加以硬件为中心的软硬件协同设计思路——针对特定问题和领域的特殊硬件架构(Domain Specific Architecture, DSA)将会展现实力,DSA的例子包括最常见的图形处理器(Graphics Processing Unit, GPU)、用于深度学习的神经网络处理器(Network Processing Unit, NPU),以及软件定义处理器(Software Defined Processor, SDP)等。DSA是一种特定领域的可编程处理器,它仍然是图灵完备的,但针对特定类别的应用进行了定制[21]。在硬件领域,DSA通常被称为加速器,因为与在通用CPU上执行整个应用程序相比,DSA可以拆分和加速应用程序的部分或全部。此外,这种加速器具备更好的性能,因为它们更贴近特定领域的计算顺序和访存模式。在特定领域的应用中,DSA加速器的硬件开销更少,速度更快,能耗更低,开发周期更短。

在深度学习的计算过程中,无论在训练阶段(Training)还是推理阶段(Inference),深度学习模型都有着巨大的计算量需求。随着不同领域数据集的不断涌现和神经网络的不断发展,更为复杂的网络模型结构也将会被提出,这种算力需求将变得更为迫切,它对硬件计算能力提出了特殊需求。因此,面向特定领域和特殊神经网络模型或算法而定制的深度学习加速器应运而生,加速器的产生为深度学习的发展带来了新的机遇和活力。

深度学习加速器是软硬件协同设计的产物,是面向软件算法而定制的硬件结构。深度学习包含训练阶段和推理阶段,GPU常用做训练阶段的加速器,而一般所说的加速器,大多指面向模型推理阶段而定制的运行时加速器。在加速器中,由于数字电子电路更擅长处理定点运算和移位运算,而一般训练好的模型权重主要是较长精度的浮点数,需要经过量化(Quantization)处理,将长精度浮点数变为短精度浮点数或定点数,这样可以大大减小推理过程的计算量,从而获得性能上的提升[22-23]。实验证明,这种量化对模型推理和识别的精度影响微小[24-26]。例如Google团队对ResNet-50网络的float point-32型权重文件进行了8-bit量化,推理速度提升3倍的同时推理精度的损失只有不到1%[22]。训练好的神经网络模型,经过权重量化,再将各层经算子拆分,最后映射到加速器上进行实时的推理和预测,该流程如图3所示。

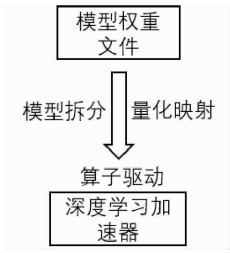

图 3　深度学习软硬件协同设计流程

Fig. 3　Software and hardware co-design of deep learning

1.3　深度学习加速器

1.3.1　加速器设计思路

由于深度学习对硬件平台的计算能力有较高需求,计算机硬件技术的进步是神经网络高速发展的强大动力,从最初的通用处理器 CPU,到后来适合大吞吐量的高性能图形处理器,再到专用集成电路(Application Specific Integrated Circuit, ASIC),以及面向特定计算顺序和访存模式而定制的现场可编程门阵列(Field Programmable Gate Array, FPGA)、CGRA(Coarse-Grained Reconfigurable Array),深度学习硬件计算平台经历了一系列的发展。图 4 列出了 CPU、GPU、ASIC 和 FPGA 4 种典型的硬件计算平台在性能、灵活性、功耗比、适用性和可编程定制性方面的能力对比。

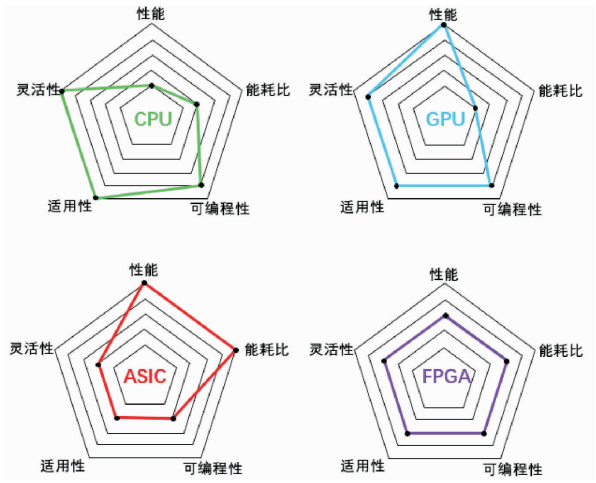

图 4　CPU、GPU、ASIC、FPGA 深度学习硬件计算平台性能对比

Fig. 4　Comparison of CPU, GPU, ASIC and FPGA

这些硬件平台为深度学习提供了强大的计算能力和访存能力的支持。一般来讲,CPU 更适用于处理指令调度灵活的通用控制任务,而 GPU 更加适用于处理有较大数据吞吐量的计算任务,比如实时动画游戏和神经网络在特定数据集下的训练。虽然专用集成电路 ASIC 在性能功耗比方面比其他硬件平台具有明显优势,但其结构一旦固化便无法修改,硬件开发周期过长,硬件一旦固化,便无法进行可编程配置,这并不适用于发展更新换代较快的深度学习神经网络的计算[27]。

相比之下,FPGA 具有面向特定领域算法、特定计算顺序和访存模式可编程定制的特点,FPGA 芯片的硬件资源可以灵活配置和分配,开发周期短,这使得 FPGA 更加适用于快速发展和灵活多变的神经网络模型的推理计算。此外,得益于近几年新兴的异构计算技术的崛起,研究人员可以将 CPU 和 FPGA 融合到单个片上系统(System on Chip, SoC),这样就可以将传统 CPU 灵活调度优势和 FPGA 强大的并行计算能力有机结合,构造出 SoC 级别的深度学习加速器。加速器系统集成在单颗芯片上可以大大降低 CPU 和 FPGA 之间的访存通讯时间,进一步提高深度学习模型的计算速度。表 1 列出了几种常见的深度学习加速器,文献[28]给出了近 10 年来几乎所有的加速器。

表 1　深度学习加速器举例

Table 1　Example of deep learning accelerators

加速器	应用阶段	设计者	年份
DianNao[29]	Inference	Chen, et al	2014
Graphcore IPU[30]	Inference/Training	Graphcore	2017
Brainwave[31]	Inference	Jeremy, et al	2018
TPU[32-33]	Training	Google	2018
NVDLA[34]	Inference	NVIDIA	2018
Equinox[35]	Inference/Training	Mario, et al	2021
Transformer-Accelerator[36]	Inference/Training	Wang, et al	2022

此外,可以将 FPGA 开发过程中成熟的且已验证正确的深度学习专用计算模块(深度学习加速器模块)打包为可移植的 IP 核(Intelligent Property Core)。这种 IP 核相当于单个硬件计算模块,在开发大型计算系统时可以被拿来直接使用。IP 核通过片上总线与 CPU 或其他计算模块直接连接,这样可以降低开发成本,缩短开发周期,便于促进深度学习加速器的开源化。目前,主要的 FPGA 生产厂家有赛灵思(Xilinx,2021 年被 AMD 公司收购)、英特尔(Intel Altera)和中国的复旦微电子等。这些厂商提供性能良好的 FPGA 芯片,为深度学习加速器的开发提供了很好的平台,在学术界和工业界备受青睐。

1.3.2　基于 FPGA 的深度学习加速器

由以上分析可知,FPGA 在构建深度学习加速器方面具有特殊优势,目前已经成为构建深度学习加速器的主流硬件平台,本小节主要介绍基于 FPGA 的深度学习加速器。

基于 FPGA 的深度学习加速器异构计算系统如图 5

所示,加速器系统主要由负责调度深度学习模型计算的CPU、负责并行计算的脉动阵列(PE array)、片上缓存、片外主存 DRAM 以及相应的接口控制器 5 大部分组成。从硬件方面来看,初期量化好的模型权重文件、模型输入数据以及负责模型计算的调度主程序,都放在内存 DRAM 当中。CPU 通过运行编译好的调度程序,将神经网络逐层拆分,利用各层对应的算子,通过逐层单层计算来复用加速器中的硬件资源。计算完成一层后缓存中间结果,然后再进行下一层的计算,直到最后一层计算完成,输出整个模型的预测结果。CPU 可以通过中断方式建立与加速器 IP 核的控制同步,当各层计算完成时加速器 IP 核发送给 CPU 一个中断信号,告诉 CPU 本次计算完成,然后输入下一个数据[37]。

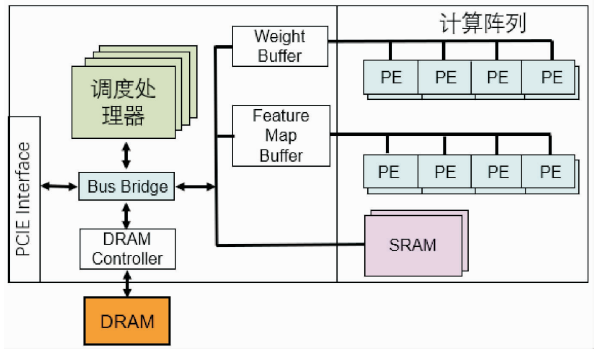

图 5 基于 FPGA 的加速器异构计算系统

Fig. 5 Heterogeneous computing system of FPGA accelerator

在软件方面,调度程序可直接通过 mmap()函数在进程虚拟内存空间的用户空间建立加速器与外设配置寄存器的直接映射,也可通过系统调用,从用户空间过度到内核空间,再通过 ioremap()函数以 IO 内存的方式访问加速器外设。主存与加速器之间的大块数据搬运可以通过系统调用函数经 CPU 转运,也可以通过 DMA 方式进行[38]。

在优化计算性能方面,可以利用多组寄存器来构造 ALU,充分利用数据级并行,构造单指令多数据流(Single Instruction Multiple Data, SIMD)的计算架构,多个 PE 阵列进行数据级并行计算。在加速器局部存储的访问上,通常利用 Ping-Pang buffer 双缓冲机制来缓解局部访存 - 计算瓶颈,以提高局部数据计算速度。在计算过程中,输入特征图(Feature map)矩阵和权重参数矩阵相乘时,可以进行分片(Tile)处理,将分片大小划分为 2^n,可以充分发挥二叉树的累加计算性能。

下面以目前深度学习中最广泛采用的卷积层(Convolution)运算为例进行论述。相对于全连接层,神经网络卷积层运算采用了卷积核滑动窗口机制,即卷积核(滤波器)对上一层输入特征图进行局部感知,输出特征图通道内权重参数共享,这 2 种机制可以大大减少权重参数的个数。

在卷积层,单个卷积核的通道个数应与输入特征图的通道数一致,而输出特征图的通道数取决于卷积核的个数。每个卷积核在输入特征图上以步幅 S 移动,每移动一次,在加速器中就进行一次乘加运算。特别地,当卷积核的大小为 1 × 1 时,卷积层退化为全连接层。图 6 画出了三通道输入特征图在 2 × 2 卷积核滑动计算的硬件映射,图中 DFF 表示 D 型触发器,用来寄存中间结果。

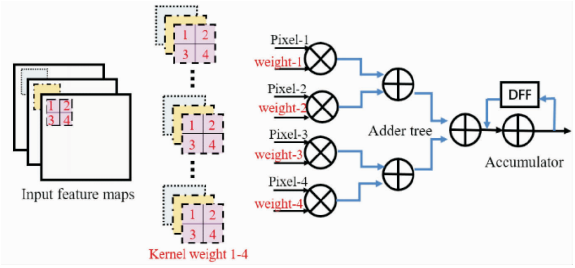

图 6 FPGA 加速器中卷积计算的硬件映射

Fig. 6 Hardware mapping for convolution computation in FPGA accelerator

卷积计算过程中由输入特征图到输出特征图计算过程的算法伪代码[39]如下:

```
for (m=0, m<Nof, m++);                    //输出通道维度 Nof
  for (r=0, r<Noy, r++);                  //输出特征图宽度 Noy
   for (c=0, c<Nox, c++);                 //输出特征图长度 Nox
    pixel(m,r,c) = pixel(m,r,c) + bias(m); //加入偏置 bias(m)
    for (n=0, n<Nif, n++);                //输入通道维度 Nif
     for (ky=0, ky<Nky, ky++);            //卷积核宽度 Nky
      for (kx=0, kx<Nkx, kx++);           //卷积核长度 Nkx
       pixel(m,r,c) + = pixel(n,r*S+ky,
        c*S+kx) * weight(m,n,kx,ky);      //输出单像素更新
```

FPGA 加速器中卷积的计算过程如图 7 所示。

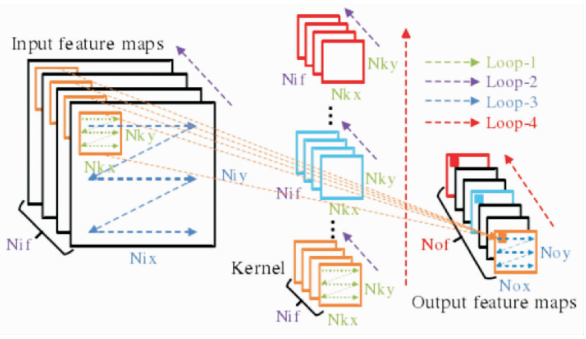

图 7 FPGA 加速器中卷积的计算过程[39]

Fig. 7 Computation process of convolution in FPGA accelerator

2　深度学习算法层面安全威胁

加速器的攻防技术主要围绕硬件进行,然而,加速器是面向深度学习模型算法而定制的专用硬件系统,其安全问题又具有领域相关性,与模型自身安全问题密不可分。因此,本部分先对深度学习模型自身的安全问题进行概括性介绍,然后详细介绍加速器系统的攻击技术及防御技术。

对于深度学习模型本身,典型的安全威胁有对抗样本攻击(Adversarial Example Attack)[40]、模型逆向攻击(Model Inversion Attack)[41-42]、模型萃取攻击(Model Extraction Attack)[43-44]、数据投毒攻击(Poisoning Attack)[45]和成员推理(Membership Inference Attacks)攻击[46]等,其中应用最多的为对抗样本攻击[47]。

模型萃取攻击,也称为模型提取攻击,是攻击者通过循环地向深度学习模型发送数据并查看模型相应的响应结果,来推测深度学习模型的参数或功能,从而复制出一个功能相似甚至完全相同的深度学习模型的攻击方法。这种攻击方法由 Tramèr 等[43]在 2016 年提出。

模型逆向攻击,该方法利用深度学习系统提供的一些 API 来获取模型的一些初步信息,并通过这些信息对模型进行逆向分析,获取模型内部的隐私数据[41]。2020 年,Zhang 等[42]借助生成对抗网络 GAN 提出了“生成式模型逆向攻击”,生成式模型逆向攻击能够更加准确地反演出被攻击模型。

数据投毒攻击是指攻击者对训练数据集进行投毒,从而影响已训练模型在未被污染的测试集上面的表现,降低模型的预测准确率。2019 年,Sun 等[45]提出针对联邦机器学习的数据投毒攻击方法:AT2FL,AT2FL 可以有效地推导出有毒数据的隐式梯度,进而计算出联邦机器学习中最优的攻击策略。

对抗样本攻击是应用最多也是最主要的安全威胁。2013 年,Szegedy 等[40]首次提出对抗样本(Adversarial Examples)的概念,其原理是攻击者通过对输入数据增加一个定制的微小扰动,可使深度学习模型形成稳定误判,以实现攻击目的。对抗样本攻击分为白盒攻击(White-Box Attack)、黑盒攻击(Black-Box Attack)和物理攻击(Physical Attack)等[48-50]。另外,根据攻击后是否预设结果,对抗样本攻击可分为有目标攻击和无目标攻击[51]。构造对抗样本的方法很多,常用经典算法有:Box-constrained L-BFGS、快速梯度下降法 FGSM、JSMA、单像素攻击、C&W、Deep Fool 和 ATNs 等[52-53],近两年研究人员还提出了基于 GAN 的对抗样本攻击[54]。

针对以上安全威胁,研究人员也提出了相应的防御技术。常用的防御方法有对抗训练(Adversarial training)、样本检测、权重枝剪(Weight pruning)、随机化(Randomization)和去噪(Denoising methods)[55]等,本文在第三部分着重讨论加速器硬件层面的安全问题。

3　深度学习加速器关键攻防技术

3.1　加速器系统攻击技术

从深度学习模型应用接口到底层加速器、存储系统,从拒绝服务(Denial-of-Service, DoS)到获取整个模型结构,深度学习加速器系统的攻击方法众多[56-58]。从大类别来讲,典型的攻击技术有侧信道攻击(Side-channel Attack)[59-61]、硬件木马攻击(Hardware Trojan Attack)[62-65]、故障注入攻击(Fault-injection Attack)[66-68],以及它们之间的融合攻击。攻击方法及基本原理如表 2 所示。

表 2　深度学习加速器系统攻击技术

Table 2　Attack techniques of deep learning accelerator system

攻击方法	基本原理
侧信道攻击	通过泄露的物理状态信息(时序、访存模式等),获取内部密钥或模型结构相关信息
硬件木马	在加速器中植入恶意模块,在特定条件下触发,以窃取 DNN 模型或破坏加速器
故障注入攻击	通过给加速器或存储器关键位置制造故障,最大程度降低模型鲁棒性或破坏系统

3.1.1　侧信道攻击

如图 8 所示,侧信道攻击(Side-channel Attack)是利用探测到的一些信道的输入和输出信息,来发掘加密系统硬件的弱点,从而绕过加密算法强有力的防护措施。具体来讲,侧信道攻击是利用硬件平台处理密码算法执行过程中泄漏的与内部运算紧密相关的多种物理状态信息,如时钟信息、声光信息、功耗、电磁辐射以及运行时间等,再结合统计学手段来进行获取秘钥相关或其他数据相关的攻击。常见的侧信道攻击实现方法包括时序侧信道攻击(Timing Side-Channel Attack)、主存测信道攻击(Memory Side-Channel Attack)、缓存侧信道攻击(Cache Side-Channel Attack)、功耗侧信道攻击(Power Side-Channel Attack)和电磁侧信道攻击(Electromagnetic Side-Channel Attack)等。

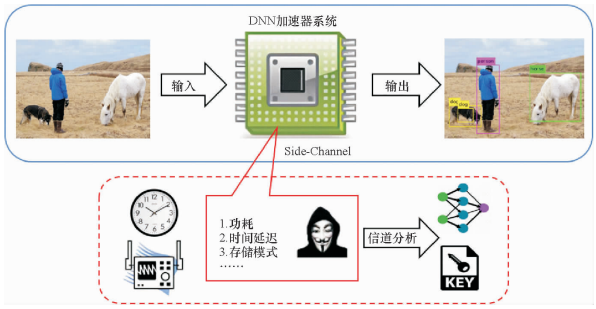

图8 深度学习加速器侧信道攻击原理

Fig. 8 Side-channel attack in deep learning accelerator

在深度学习加速器中,攻击者可以利用侧信道泄露的信息对深度学习模型的信息进行推断,获得网络层级结构,或者获取加速器中深度学习模型的推理规律,以实现盗取模型信息或根据该信息对推理过程或推理结果实施破坏的目的。

Hua 等[69]利用逆向工程,向执行 8 层 AlexNet 和 18 层 SqueezeNet 模型的加速器中进行试探性输入,然后观测片外存储的 Memory Pattern 和 Timing 侧信道信息,进而可以推测出潜在的神经网络结构和零枝剪情况下的模型权重信息,该方法可以绕过强大的加密防御技术。

Jha 等[70]提出一种两阶段攻击方法"DeepPeep",该方法利用对目标加速器时延、访存足迹和输入每帧图像后的功耗等侧信道信息的探测,并与已知加速器的表现进行对比,可用来推测由 Dense Block、Channel Shuffling 和 Depthwise CONV 等模块堆叠而组成的 DNN 模型结构。

针对公用或租用 GPU 可以被多个用户在细粒度级别共享,Naghibijouybari 等[59]利用 OpenGL 发送任务到 GPU,这样攻击者和受害者就可以交错使用共享的 GPU,通过线程级并行的上下文信息,攻击者可以通过存储大小和时间侧信道结合 CUDA 的资源跟踪 API,来推测受害者的 DNN 模型结构。

Yan 等[61]提出利用 Cache 侧信道的"Cache Telepathy",该方法基于 DNN 推理严格依赖于广义矩阵相乘 GEMM 这一规律,根据 GEMM 的 Cache 测信道信息可以获得 DNN 结构参数信息,实验表明,该方法可以有效攻击 VGG-16 和 ResNet-50 模型。

3.1.2 硬件木马攻击

木马攻击包含软件木马攻击和硬件木马攻击,硬件木马(Hardware Trojan)是指硬件设计者或第三方 IP 核供应商故意留下的恶意电路功能模块。从功能上来讲,硬件木马包含触发模块(Trigger)和实施攻击的负载模块(Payload)2 个部分,Trigger 实时监测硬件执行环境,可以在特定条件下触发 Payload 负载,Payload 用来实施攻击。硬件木马中这些模块一般潜伏在正常的硬件电路之中,仅在特定条件(通常是攻击者设计的小概率事件)下被触发。Trigger 一旦触发,Payload 模块能够被攻击者利用,进而对深度学习加速器进行有目的的修改,使原始硬件平台出现功能规范之外的不期望行为,这些行为可以在硬件层面控制或泄露深度学习模型的数据信息,甚至直接损坏深度学习加速器,使之不能输出正确结果。

对于硬件木马攻击神经网络加速器的研究,2018 年,Clements 等[71]基于 JSMA 算法在卷积神经网络加速器中植入了硬件木马,并成功对 MNIST 和 CIFAR-10 数据集进行了攻击。同年,Ye 等[72]提出在深度学习模型输入的图像上加上添加 0.000 356% 的修改便可以 100% 激活藏在加速器平台上硬件木马,从而可以操纵神经网络模型的预测结果,该硬件木马只占原始加速器硬件资源开销的 0.005 1%,难以被用户察觉。

2019 年,Zhao 等[65]在内存控制器 memory controller 中注入硬件木马,通过监测读写存储器的泛洪随时间的变化时序规律,来推测输入数据信息。他们利用一种特殊的分形图像作为输入来触发硬件木马,可以直接控制深度学习加速器的输出结果。

2020 年,Liu 等[62]在干净的 AI 加速器中加入了一个硬件木马旁路模块,将图像识别中正常的图像以特定的序列输入到神经网络加速器,通过旁路模块的 PRB 模块来检测该输入序列,当旁路模块检测到事先设定好的特定的输入序列后,可以触发该硬件木马,该方法可以用来躲避深度学习输入图像预处理防御技术。

3.1.3 故障注入攻击

故障注入攻击(Fault-injection Attacks)又称为错误注入攻击,是指在硬件层面对深度学习加速器引入几乎察觉不到的微小的故障或错误,从而干扰深度学习的正确计算。

在深度学习加速器中,层内噪声注入(Noise-injection)[68]和关键位反转攻击(Bit-flip attack)[66,73]是目前比较成熟的故障注入攻击。常用的故障注入攻击是针对深度学习模型的权重(Weight)和偏置(Bias)参数而开展的 Bit-Flip Attack 攻击(简称 BFA)。类似于黑盒攻击中的单像素攻击寻找关键攻击像素点的方法,BFA 攻击利用特定方法寻找到对 DNN 模型鲁棒性影响最大的关键权重点进行硬件或物理层面的攻击,攻击后可使模型预测精确度严重降低,甚至使加速器平台无法正常进

行推理。常见的 BFA 攻击实现方法有行锤(row-hammer)、时钟脉冲故障(clock glitch)和电磁类辐射故障(electromagnetic radiation)等。

Rakin 等[66-67]提出“关键位反转攻击”,他们利用损失函数的梯度来寻找权重中对深度学习模型预测精确度影响最大的脆弱位(vulnerable bits),进而实施攻击。他们发现,对于 AlexNet、ResNet-18 和 ResNet-50 模型,分别需要翻转 17 位、13 位和 11 位关键位就可以将模型平均推理精度降低到 0.2%。作为对照实验,随机性的翻转权重的 100 位对 DNN 模型准确度却基本没有影响。此外,对照实验表明,权重枝剪(Weight Pruning)和对抗训练(Adversarial Training)也不能有效防御该方法的攻击。

Liu 等[68]利用噪声注入的方法,评估了 10 种 DNN 模型对内层注入噪声的敏感性。他们发现对于传统的卷积网络(AlexNet 和 VGG-16),最后一个卷积层是最易受攻击的。对最后一个卷积层添加 0.3% 的噪声,就可以使模型分类精度下降 46.3%(AlexNet)和 36.3%(VGG-16)。而对于先进的网络结构(Inception, ResNet 和 DenseNet),改变模型权重的 0.1%,或者改变每个通道维度一个位的扰动,就可以改变整个模型的原始预测结果。

3.2　加速器系统防御技术

针对深度学习加速器系统的以上攻击,以及模型自身层面的其他安全威胁,研究人员也提出了硬件加速器层面的防御方法。表 3 列出了深度学习加速器系统相关的关键防御技术及其基本原理。

表 3　深度学习加速器系统防御技术

Table 3　Defense techniques of deep learning accelerator system

防御方法	基本原理
加密与认证	加密保证隐私,认证保证合法
检测与监测	检测潜在漏洞,监测执行过程中的威胁
可信与隔离	创造可信执行环境,隔离关键敏感区域
诱骗与干扰	设置陷阱和干扰,使攻击无法进行
枝剪与量化	优化模型结构,避开在模型层面攻击
随机化	增加随机区域,攻击者难以捕获有效信息

3.2.1　加密与认证

加密(Encryption)与认证(Authentication)是最为经典的安全技术。如图 9 所示,信息传输过程中的安全威胁主要分为 4 大类:窃听、篡改、伪装和否认,与其对应的安全特性分别为:机密性(Confidentiality)、完整性(Integrity)、认证(Authentication)和不可否认性(Non-repudiation)。具体对应的防御方法分别是加解密、单向散列函数、认证和数字签名。一般来讲,对于深度学习加速器系统,不存在不可否认性这一安全问题。

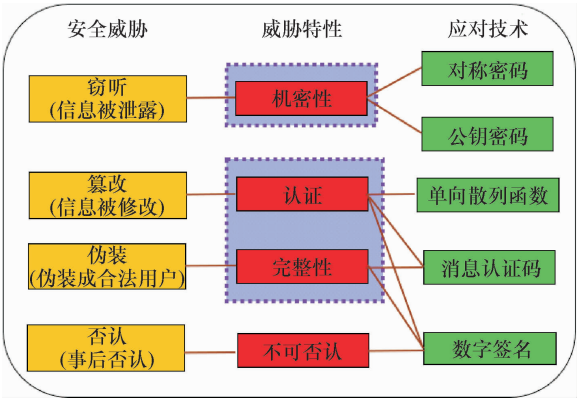

图 9　安全分析之密码学家的工具箱

Fig. 9　Cryptographer's toolbox for security analysis

2019 年,Wang 等[74]针对 Model Inversion Attack 提出深度学习安全加速器架构 NPUFort。他们利用 AES 加密算法的 CTR 模式对片上加速器的指令流、数据流和控制流的关键总线路径进行加解密,用来防御模型逆向攻击。该加速器也可以从一定程度上防御侧信道攻击。

Guo 等[75]提出基于物理不可克隆函数(Physical Unclonable Functions, PUF)进行模型认证,来保护深度学习模型参数,PUF 混淆后的 DNN 模型参数只能在特定硬件设备上运行。在 AlexNet 上的实验表明,仅需 0.44% 的额外硬件开销,便使攻击者的攻击精度下降到 1.2%,而模型性能损失仅为 0.83%[76]。

为了保证深度学习计算平台数据的机密性和完整性,Hua 等[77]提出安全的 DNN 加速器架构 GuardNN。他们对利用 AES 对称密码和消息认证码(Message Authentication Code, MAC)机制实现的 FPGA 加速器原型系统进行评估,发现加入安全机制后对深度学习推理性能的影响低于 2%。

致力于实现实时加密态 DNN 计算,2021 年,Reagen 等[78]提出基于全同态加密(Homomorphic Encryption)的深度学习加速器优化策略 Cheetah,他们在一个 5 nm 特殊加速器中将 ResNet-50 模型的密态计算速度提升到了实时推理。

3.2.2　检测与监测

检测(Detection)是指对深度学习加速器的输入数据,或者内部硬件和模型参数进行潜在安全威胁和漏洞的检查,以排除诸如对抗样本和软硬件木马等的威胁。监测(Monitoring)是指在深度学习模型计算过程中增加旁路或辅助监视功能模块,进行实时监测,以排除加速

器系统运行时内部或外部出现的各种安全威胁。

2019年,Reshma等[79]提出用深度学习的方法来检测网表中的硬件木马。对于一个给定的电路,他们利用深度学习和K-means聚类方法来提取网表中一种称之为"可控迁移几率"的值作为木马标记,该方法在小于6 s的时间内对ISCAS'85 benchmark电路中木马感染节点的检测准确率为100%。

2020年,Wang等[80]提出基于RISC-V的弹性异构深度学习加速器SoC架构DNNGuard,该加速器可以并行、高效地支持2个并行神经网络的计算:①推理神经网络的计算;②检测对抗样本攻击的神经网络的计算。为了降低片外存储器的访存开销和提高硬件资源利用率,DNNGuard架构还引入动态资源调度机制并扩展了AI指令集,在NVDLA开源加速器上的实验表明,该调度方案可以比Base line加速器实现1.42倍性能的提升。

2021年,Sharma等[81]提出基于部分逆向工程(Partial Reverse Engineering, PRE)的改进方法,利用深度卷积神经网络(DCNN)来检测IC布局版图中的硬件木马,该方法在Trust-Hub和ISCAS数据集上对硬件木马的检测精度达到了99%和97%。

3.2.3 可信与隔离

在硬件上建立可信操作区域,对隐私计算进行隔离,常用到可信执行环境(Trusted Execution Environments, TEE)的概念。如图10所示,TEE的地位相当于一个小的操作系统管理区,TEE允许应用程序在私有区域处理敏感数据,该区域也常被称作"飞地",英文译为enclave。常见的TEE技术有ARM TrustZone技术、Intel的软件保护扩展SGX技术等。TEE技术常被用到深度学习加速器安全防护当中。

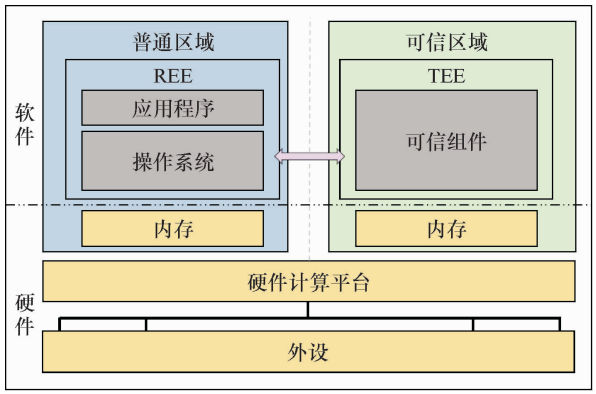

图10 可信执行环境TEE

Fig. 10 Trusted execution environment

为保证资源有限的移动端用户中DNN模型的计算速度和安全,Gangal等[82]提出基于ARM TrustZone和SGX的移动可信执行环境HybridTEE。HybridTEE利用"基于目标识别"和"尺度不变特征变换SIFT"2种隐私度量技术,将DNN模型的计算进行隐私感知拆分,将一部分计算送到服务器,并在移动端和服务器之间建立安全传输通道,来保证移动端的计算速度和安全。安全和性能评估实验表明,HybridTEE在保证安全的前提下可将Darknet-19和GoogleNet的推理速度分别提升1.75倍和3.5倍。

为了支持深度学习云计算平台和诸如GPU等板卡级系统中的大规模数据隐私计算,Zhu等[83]提出板卡级异构可信执行环境HETEE。他们在PCIe总线网上设置HETEE Box作为安全隔离区,利用资源池技术动态划分计算任务,并利用一个TCB栈来进行安全管理。他们在搭建的HETEE安全硬件平台上进行了ResNet-152的安全训练和推理实验,实验表明,安全训练和推理过程中的访存开销比一般情况分别降低了0.95% 和2.17%。

2022年,Lee等[84]提出神经网络加速器可信执行环境TNPU,他们利用一种称之为"无树完整性保护(tree-less integrity protection)"的机制,该机制给CPU enclave中每个计算张量维护一个张量版本号,通过追踪带版本号的张量更新的数据流,可以有效地验证张量中数据的"近代性",从而可以保护数据的完整性。该方法可以避免传统的内存硬件保护机制中hash cache不命中而带来的性能损失,在三星Exynos 990和ARM Ethos N77中的实验表明,单个NPU的性能平均提高了10.0%和7.5%。

3.2.4 诱骗与干扰

诱骗与干扰是指在加速器的数据访存过程中,增加一些混淆性的访存模式,或在常规访存、计算规律中加入一些干扰,从而使攻击者得不到透明有效的规律性信息,同时又不带来繁重的性能开销。在深度学习加速器中,神经网络不同层的数据访问模式是不同的,攻击者可以通过监测存储访问模式的规律,并和已知网络的访存模式进行对比,从而推测出模型的内部层间结构。常用的诱骗干扰技术有"不经意随机访问(Oblivious RAM, ORAM)"技术,它通过混淆每一次访问过程,使其与随机访问不可区分,从而保护数据的访问操作、访问位置等信息。

在深度学习加速器中,为了防止攻击者利用内存访问模式侧信道对DNN模型进行逆向攻击,Liu等[85]提出了一种防御性的内存访问模式,该模式主要基于3种技术:随机打乱(Oblivious shuffle)、地址空间布局随机化(Address space layout randomization, ASLR)和伪内存访问模式(Dummy memory access, DumMA),该方法带来

的内存访问额外开销极低,且不会随着 DNN 网络深度的增加而增加。

为了保护深度学习模型权重文件,Zhao 等[86]提出基于错误掩码(Error mask)的神经网络加速器设计方案 AEP,他们故意制造少量 DRAM 时序错误单元并产生相应错误掩码,这些错误单元的分布只依赖于特定设备时序,并且只能由错误掩码来进行屏蔽。然后,在错误掩码的过滤下进行模型权重的训练,得到的权重文件只能在本地执行,即使攻击者窃取训练的权重文件,也不能有效运行。实验表明,通过注入 0.1% ~5.0% 错误,AEP 带来的本地精度损失可以忽略不计,但在没有错误掩码的平台精度损失到基本不能使用。此外,研究人员还通过 Cache 分区[87]和 Cache 锁存[88]等方法来干扰通过 Cache 侧信道攻击访存数据。

3.2.5　枝剪与量化

针对对抗样本攻击和故障注入等攻击,研究人员还提出了模型权重层面的枝剪(Pruning)或量化(Quantization)方法进行防御。通过权重的量化或枝剪,可以从模型权重上增强 DNN 鲁棒性和抗攻击性。

为了防御关键位反转攻击(BFA),Li 等[89]提出一种权重重构的方法,具体来讲,在推理过程中对权重进行一种“平均 - 量化 - 裁剪”的 3 步重构操作,BFA 引起的权值扰动将会被最小化或扩散到相邻权重去,该方法可以提升 DNN 模型在基于梯度的 BFA 攻击下的鲁棒性。实验表明,在 5 轮 BFA 迭代攻击下被保护的 ResNet-18 在 ImageNet 识别精度保持在 60% 以上,而没有保护机制的模型攻击后识别精度下降到了不足 1%。

针对 Bit-Flip 权重攻击,He 等[90]系统地分析了 Bit-Flip 攻击原理,他们发现常规的权重枝剪和对抗训练并不能有效防御权重 Bit-Flip 攻击。基于此,他们提出一种利用“权重二值量化感知训练”加“分段聚类约束”的防御对策,在 ResNet-20 和 VGG-11 模型上的实验表明,Bit-Flip 要实现相同的攻击效果(如对 CIFAR-10 数据集的识别精度低于 11%),该种防御情况下的模型的位翻转数量分别需要比无防御情况下提升 19.3 倍和 480.1 倍。

Rakin 等[91]提出用“动态量化激活 DQA”的方法来提高 DNN 模型的鲁棒性,他们将激活函数量化的位宽值作为一个调优参数进行对抗训练,对抗训练过程中该参数的调整变化可以在每个激活层抑制和过滤对抗噪声,对抗学习后的量化宽度可以用来提高模型的鲁棒性,从而防御对抗样本攻击。LeNet 网络对 MNIST 以及 Resnet-18 网络对 CIFAR10 数据集上的实验表明,该量化方法在减小计算量的同时,也提高了模型的鲁棒性。

3.2.6　随机化

随机化是指在访存、生成密钥或构造其他数据结构时对操作对象进行随机化处理,使攻击者捕获不到有规律的对象信息,进而达到防御相应攻击的目的。在深度学习加速器系统的防御技术当中,随机化技术与前面提到的诱骗、干扰技术有重合之处,然而随机化这种叫法一直是计算机防御领域里的一种惯例。

2021 年,Fu 等[92]提出具有对抗鲁棒性的加速器软硬件协同设计框架“2-in-1 Accelerator”,他们利用一种随机精度开关算法(Random Precision Switch, RPS),在训练和推理过程中将深度学习模型进行随机精度的定点量化,然后映射到加速器中,在 6 种 DNN 模型和 4 种数据集上的实验表明,该方法可以将 DNN 模型遭受对抗攻击时的鲁棒精度提高 24.48%。

Liu 等[85]在保护 DNN 模型计算时内存访问模式免遭侧信道攻击时,也用了地址空间布局随机化的技术。

4　软硬件协同防御

以往软硬件分离的设计思路中,硬件设计过程中缺乏对软件构架和实现机制的清晰了解,在系统安全设防时,由于受到设计空间的限制,只能改善硬件/软件各自的安全性能,不能对系统进行综合优化,很难充分利用硬软件资源的交互,难以适应面向特定算法而定制的异构计算系统。软硬件协同设计是使软件设计和硬件设计作为一个有机的整体进行并行设计,软硬件互相影响和配合,在更细粒度级别实现最佳结合,从而使系统获得更高的工作性能、安全性能。由于深度学习加速器是面向特定领域或算法而定制的硬件计算系统,软硬件协同是其与生俱来的特点。深度学习加速器系统软硬件协同防御的路线有 2 层含义:①用硬件安全机制防御深度学习输入样本、模型权重或训练推理算法过程层面的攻击;②用优化的模型权重、输入数据或训练推理过程来防御硬件层面的攻击。

随着 EDA 技术的发展,深度学习加速器设计人员可以利用高层次综合(High-level Synthesis, HLS)工具,使用面向对象的高级语言(例如 C ++、Scala、Chisel 和 PyGears 等敏捷开发语言)来设计针对深度学习模型特定并行计算规模和访存结构的加速器硬件。

伴随着诸如全同态加密计算、Transformer 加速器、联邦学习和云加速平台等计算机技术的不断进步和神经网络结构的不断发展,许多新的安全威胁也会不断涌现。追求安全和高效的深度学习系统,始终是软硬件研

究人员不断努力探索的目标。

5 结束语

离开硬件谈AI安全是追求无源之水。在硬件领域,研究人员主要聚焦于深度学习加速器在特定领域推理的精度、速度以及访存带宽性能的提升,而往往忽略其安全问题。而深度学习的安全领域当中,以往研究人员更多地关注深度学习模型本身的鲁棒性以及其训练、推理过程中算法层的安全问题,而针对其硬件计算平台——深度学习加速器的安全关注相对较少。尤其在后摩尔定律时代,随着面向特定神经网络结构或特定模型算法而定制的专用异构计算系统,以及神经网络芯片NPU的崛起,深度学习硬件加速器系统的安全问题更加凸显。

本文从硬件角度入手,系统地分析了深度学习加速器系统的关键攻击及防御技术,以求帮助科研人员或工程技术人员快速了解本领域,为软硬件协同技术奠定基础。

参考文献:

[1] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521: 436-444.

[2] Vargas R, Ruiz L. Deep learning: Previous and present applications[J]. Journal of Awareness, 2017, 2:11-20.

[3] Muhammad K, Ullah A, Lloret J, et al. Deep learning for safe autonomous driving: Current challenges and future directions [J]. IEEE Transactions on Intelligent Transportation Systems, 2020,22(7):4316-4336.

[4] Milz S, Arbeiter G, Witt C, et al. Visual SLAM for automated driving: Exploring the applications of deep learning[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Piscataway: IEEE, 2018: 360-370.

[5] Li D, Dong Y. Deep learning: Methods and applications[J]. Foundations and Trends in Signal Processing, 2014, 7:197-387.

[6] McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity[J]. Bulletin of Mathematical Biophysics, 1943, 5: 115-133.

[7] LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural Computation, 1989, 1(4):541-551.

[8] Liu P P, Qiu X P, Huang X J. Recurrent neural network for text classification with multi-task learning[EB/OL]. (2016-05-17)[2022-03-15]. https://arxiv.org/pdf/1605.05101.pdf.

[9] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2016: 770-778.

[10] Zhou J, Cui G Q, Hu S D, et al. Graph neural networks: A review of methods and applications[EB/OL]. (2018-12-25)[2022-03-15]. https://arxiv.org/pdf/1812.08434.pdf.

[11] Vaswani A, Shazeer N, Parmar N, et al. Attention is all your need[EB/OL]. (2017-06-21)[2022-03-15]. https://arxiv.org/pdf/1706.03762.pdf.

[12] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. Piscataway: IEEE, 2016: 779-788.

[13] Howard A G, Zhu M L, Chen B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications [EB/OL]. (2017-04-01)[2022-03-15]. https://arxiv.org/pdf/1704.04861.pdf.

[14] Abbas Q, Ibrahim M, Jaffar A. A comprehensive review of recent advances on deep vision systems[J]. Artificial Intelligence Review, 2019, 52:39-76.

[15] Ossama A H, Li D, Dong Y. Exploring convolutional neural network structures and optimization techniques for speech recognition[C]//Inter Speech. Lyon: ACL,2013: 1173-1175.

[16] Tu F B, Wu Z H, Wang Y Q, et al. A 28nm 15.59μJ/token full-digital bitline-transpose CIM-based sparse transformer accelerator with pipeline/parallel reconfigurable modes[C]//ISSCC2022. Piscataway:IEEE, 2022.

[17] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Neural Information Processing Systems, 2017, 60(6): 84-90.

[18] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//The 27th International Conference on Neural Information Processing Systems. Montreal:MIT Press, 2014: 2672-2680.

[19] Li P, Che C. SeMo-YOLO: A multiscale object detection network in satellite remote sensing images[C]//2021 Internation-

al Joint Conference on Neural Networks (IJCNN). Piscataway:IEEE, 2021: 1-8.
[20] Google. Federated learning[EB/OL]. (2017-04-01)[2018-08-25]. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.
[21] Hennessy J, Patterson D. A new golden age for computer architecture[J]. Communications of the ACM, 2019, 62(2): 48-60.
[22] Krishnamoorthi R. Quantizing deep convolutional networks for efficient inference: A white paper[EB/OL]. (2018-06-01)[2022-03-15]. https://arxiv.org/abs/1806.08342.
[23] Jacob B, Kligys S, Chen B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2018: 2704-2717.
[24] Cheng G, Lu Y, Xie K P, et al. Elastic significant bit quantization and acceleration for deep neural networks[C]//IEEE Transactions on Parallel and Distributed Systems. Piscataway: IEEE,2021:1.
[25] Xie H W, Song Y F, Cai L, et al. Overflow aware quantization: Accelerating neural network inference by Low-bit multiply-accumulate operations[C]// Proceedings of the Twenty-ninth International Joint Conference on Artificial Intelligence (IJCAI2020). Yokohama: Morgan Kaufmann, 2020: 868-875.
[26] Liang T L, Glossner J, Wang L, et al. Pruning and quantization for deep neural network acceleration: A Survey[J]. Neuro Computing, 2021: 370-403.
[27] 吴艳霞,梁楷,刘颖,等. 深度学习 FPGA 加速器的进展与趋势[J]. 计算机学报,2019, 42(11):2461-2480.
[28] Tu F B. Neural networks on silicon[EB/OL]. (2017-04-01)[2022-03-15]. https://github.com/fengbintu/Neural-Networks-on-Silicon.
[29] Chen T S, Du Z D, Sun N H, et al. DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning[J]. ACM SIGARCH Computer Architecture News, 2014, 42: 269-284.
[30] Graphcore. Introduction to the IPU architecture[EB/OL]. (2019-08-06)[2022-03-15]. https://www.graphcore.ai/.
[31] Fowers J, Ovtcharov K, Papamichael M, et al. A configurable cloud-scale DNN processor for real-time AI[C]//Proceedings of the 45th International Symposium on Computer Architecture (ISCA). Piscataway: IEEE, 2018:1-14.
[32] Google. Cloud TPU[EB/OL]. (2018-01-31)[2022-03-15]. https://cloud.google.com/tpu.
[33] Google. Tearing apart google's TPU 3.0 AI coprocessor[EB/OL]. (2018-05-15)[2022-03-15]. https://www.nextplatform.com/2018/05/10/tearing-apart-googles-tpu-3-0-ai-coprocessor.
[34] NVIDIA. Hardware architectural specification[EB/OL]. (2018-06-01)[2022-03-15]. http://nvdla.org/hw/v1/hwarch.html.
[35] Drumond M, Coulon L, Pouhabibi A, et al. Equinox: Training for free on a custom inference accelerator[C]//MICRO'21: 54th Annual IEEE/ACM International Symposium on Microarchitecture. New York:ACM, 2021: 421-433.
[36] Yang W, Qin Y B, Deng D Z, et al. A 28nm 27.5TOPS/W approximate-computing-based transformer processor with asymptotic sparsity speculating and out-of-order computing[C]// International Solid-State Circuits Conference 2022, ISSCC2022. Piscataway:IEEE,2022.
[37] 梁爽. 可重构神经网络加速器设计关键技术研究[D]. 北京: 清华大学, 2017.
[38] 余奇. 基于 FPGA 的深度学习加速器设计与实现[D]. 合肥: 中国科学技术大学, 2016.
[39] 陈辰. 基于 FPGA 的神经网络加速器的规模可伸缩性研究[D]. 无锡: 江南大学, 2019.
[40] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks[EB/OL]. (2013-12-15)[2022-03-15]. https://arxiv.org/pdf/1312.06199.pdf.
[41] Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures[C]//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2015: 1322-1333.
[42] Zhang Y H, Jia R X, Pei H Z, et al. The secret revealer: Generative model-inversion attacks against deep neural networks[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE,2020: 250-258.
[43] Tramèr F, Zhang F, Juels A, et al. Stealing machine learning models via prediction APIs[C]//In 25th USENIX Security Symposium, USENIX Security 16. Austin: USENIX Association, 2016: 601-618.
[44] Hu X, Liang L, Deng L, et al. Neural network model extraction attacks in edge devices by hearing architectural hints[EB/OL]. (2019-03-01)[2022-03-15]. https://arxiv.org/pdf/1903.03916.pdf.

[45] Sun G, Cong Y H, Dong J H, et al. Data poisoning attacks against federated learning systems[EB/OL]. (2019-03-01)[2022-03-15]. https://arxiv.org/pdf/2004.10020.pdf.

[46] Shokri R, Stronati M, Song C, et al. Membership inference attacks against machine learning models[C]//2017 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE, 2017: 3-18.

[47] He Y Z, Meng G Z, Chen K, et al. Towards privacy and security of deep learning systems: A survey[EB/OL]. (2019-11-01)[2022-03-15]. https://arxix.org/pdf/1911.12562vl.pdf.

[48] Akhtar N, Mian A. Threat of adversarial attacks on deep learning in computer vision: A survey[EB/OL]. (2019-03-01)[2022-03-15]. https://arxiv.org/pdf/1801.00553.pdf.

[49] Gil Y, Chai Y, Gorodissky O, et al. White-to-black: Efficient distillation of black-box adversarial attacks[C]//North American Chapter of the Association for Computational linguistics. Minneaplis:NAACL, 2019: 1373-1379.

[50] Odena A, Dumoulin V, Olah C. Deconvolution and checkerboard artifacts[EB/OL]. (2016-03-01)[2022-03-15]. https://distill.pub/2016/deconv-checkerboard/.

[51] Yuan X Y, He P, Zhu Q L, et al. Adversarial examples: Attacks and defenses for deep learning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(9): 2805-2824.

[52] Baluja S, Fischer I. Adversarial transformation networks: Learning to generate adversarial examples[EB/OL]. (2017-03-01)[2022-03-15]. https://arxiv.org/pdf/1703.09387.pdf.

[53] Shen W J, Wan J, Chen Z Y. MuNN: Mutation analysis of neural networks[C]//2018 IEEE International Conference Software Quality Reliability and Security Companion (QRS-C). Piscataway: IEEE, 2018:108-115.

[54] Xiao C W, Li B, Zhu J Y, et al. Generating adversarial examples with adversarial networks[C]//ICLR 2018. Vancouver: ICLR, 2018: 3905-3911.

[55] Ren K, Zheng T H, Qin Z, et al. Adversarial attacks and defenses in deep learning[J]. Engineering, 2020,6(3): 346-360.

[56] Sparsh M, Himanshi G, Srishti S. A survey on hardware security of DNN models and accelerators[J]. Journal of Systems Architecture, 2021, 117: 1-30.

[57] Xu Q, Arafin M T, Guang Q. Security of neural networks from hardware perspective: A survey and Beyond[C]//2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC). Piscataway: IEEE,2021: 449-454.

[58] Isakov M, Gadepally V, Gettings K M, et al. Survey of attacks and defenses on edge-deployed neural networks[C]//2019 IEEE High Performance Extreme Computing Conference (HPEC). Piscataway: IEEE, 2019: 1-8.

[59] Naghibijouybari H, Neupane A, Qian Z Y, et al. Rendered insecure: GPU side channel attacks are practical[C]//ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2018: 2139-2153.

[60] Hong S Y, Davinroy M, Kaya Y, et al. Security analysis of deep neural networks operating in the presence of cache side-channel attacks[EB/OL]. (2018-10-01)[2022-03-15]. https://arxiv.org/pdf/1810.03487.pdf.

[61] Yan M J, Fletcher C W, Torrellas J. Cache telepathy: Leveraging shared resource attacks to learn dnn architectures[C]//USENIX Security. Boston: USENIX Association, 2020:2003-2020.

[62] Liu Z Z, Ye J, Hu X, et al. Sequence triggered hardware trojan in neural network accelerator[C]//IEEE 38th VLSI Test Symposium (VTS). Piscataway: IEEE, 2020: 1-6.

[63] Liu T, Wen W J, Jin Y. SIN2: Stealth infection on neural network—a low-cost agile neural trojan attack methodology[C]//IEEE International Symposium on Hardware Oriented Security and Trust (HOST). Piscataway: IEEE, 2018: 227-230.

[64] Li W S, Yu J C, Ning X F, et al. Hu-fu: Hardware and software collaborative attack framework against neural networks[C]//2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI). Piscataway: IEEE, 2018: 482-487.

[65] Zhao Y, Hu X, Li S W, et al. Memory trojan attack on neural network accelerators[C]//DATE2019. Piscataway: IEEE, 2019: 1415-1420.

[66] Rakin A S, He Z Z, Li J T, et al. Bit-Flip attack: Crushing neural network with progressive bit search[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE, 2019: 1211-1220.

[67] Rakin A S, He Z, Li J, et al. T-BFA: Targeted bit-flip adversarial weight attack[C]//IEEE Transactions on Pattern Analysis and Machine Intelligence. Piscataway: IEEE, 2021: 1-12.

[68] Liu W Y, Wang S, Chang C H. Vulnerability analysis on noise-injection based hardware attack on deep neural networks[C]//Asian Hardware Oriented Security and Trust Symposium (AsianHOST). Piscataway: IEEE, 2019: 1-6.

[69] Hua W Z, Zhang Z R, Suh G E. Reverse engineering convolutional neural networks through side-channel information leaks

[C]//Design Automation Conference (DAC). Piscataway: IEEE, 2018: 1-6.
[70] Jha N K, Mittal S, Kumar B, et al. DeepPeep: Exploiting design ramifications to decipher the architecture of compact DNNs[J]. ACM Journal of Emerging Trends in Computing, 2020, 17(1): 1-25.
[71] Clements J, Lao Y J. Hardware trojan attacks on neural networks[EB/OL]. (2018-06-01)[2022-03-15]. https://arxiv.org/pdf/1806.05768.pdf.
[72] Ye J, Hu Y, Li X. Hardware Trojan in CNN FPGA accelerator[C]//IEEE 27th Asian Test Symposium (ATS). Piscataway: IEEE, 2018: 68-73.
[73] Venceslai V, Marchisio A, Alouani I, et al. NeuroAttack: Undermining spiking neural networks security through externally triggered bit-flips[C]//2020 International Joint Conference on Neural Networks (IJCNN). Piscataway: IEEE, 2020: 1-8.
[74] Wang X B, Hou R, Zhu Y, et al. NPUFort: A secure architecture of DNN accelerator against model inversion attack[C]//CF2019. New York: ACM, 2019: 190-196.
[75] Guo Q L, Ye J, Gong Y, et al. PUF based pay-per-device scheme for IP protection of CNN model[C]//ATS2018. Piscataway: IEEE, 2018: 115-120.
[76] 郭青丽. 面向神经网络对抗样本的软硬件防护方法研究[D]. 北京: 中国科学院大学, 2020.
[77] Hua W Z, Umar M, Zhang Z R, et al. GuardNN: Secure DNN accelerator for privacy-preserving deep learning[EB/OL]. (2020-08-01)[2022-03-15]. https://arxiv.org/pdf/2008.11632.pdf.
[78] Reagen B D, Choi W S, Ko Y, et al. Cheetah: Optimizing and accelerating homomorphic encryption for private inference[C]//2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA). Piscataway: IEEE, 2021: 26-39.
[79] Reshma K, Priyatharishini M, Devi M N. Hardware trojan detection using deep learning technique[J]. Advances in Intelligent Systems and Computing, 2019, 898: 671-680.
[80] Wang X B, Hou R, Zhao B Y, et al. DNNGuard: An elastic heterogeneous DNN accelerator architecture against adversarial attacks[C]//ASPLOS2020. Piscataway: IEEE, 2020: 19-34.
[81] Sharma R, Rathor V S, Sharma G K, et al. A new hardware trojan detection technique using deep convolutional neural network[J]. Integration, 2021,79: 1-11.
[82] Gangal A, Ye M M, Wei S. HybridTEE: Secure mobile DNN execution using hybrid trusted execution environment[C]//2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST). Piscataway: IEEE, 2020: 1-6.
[83] Zhu J P, Hou R, Wang X F, et al. Enabling rack-scale confidential computing using heterogeneous trusted execution environment[C]//IEEE Symposium on Security and Privacy (S&P). Piscataway: IEEE, 2020: 1450-1465.
[84] Lee S H, Kim J W, Na S L, et al. TNPU: Supporting trusted execution with tree-less integrity protection for neural processing unit[C]//HPCA2022. Piscataway:IEEE,2022.
[85] Liu Y T, Dachman-Soled D, Srivastava A. Mitigating reverse engineering attacks on deep neural networks[C]//IEEE Computer Society Annual Symposium on VLSI (ISVLSI). Piscataway: IEEE, 2019: 657-662.
[86] Zhao L, Zhang Y, Yang J. AEP: An error-bearing neural network accelerator for energy efficiency and model protection[C]//International Conference on Computer-Aided Design (ICCAD). Piscataway: IEEE, 2017: 1047-1053.
[87] Mittal S. A survey of techniques for cache partitioning in multicore processors[J]. ACM Computing Surveys, 2017, 50(2): 27-39.
[88] Mittal S. A survey of techniques for cache locking[J]. ACM Transactions on Design Automation of Electronic Systems (TODAES), 2016, 21(3): 1-49.
[89] Li J T, Rakin A S, Xiong Y, et al. Defending bit-flip attack through dnn weight reconstruction[C]//Design Automation Conference (DAC). New York: ACM,2020: 1-6.
[90] He Z Z, Rakin A S, Li J T, et al. Defending and harnessing the bit-flip based adversarial weight attack[C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New York: ACM, 2020: 14083-14091.
[91] Rakin A S, Yi J, Gong B, et al. Defend deep neural networks against adversarial examples via fixed and dynamic quantized activation functions[EB/OL]. (2018-07-01)[2022-03-15]. https://arxiv.org/pdf/1807.06714.pdf.
[92] Fu Y G, Zhao Y, Yu Q, et al. 2-in-1 accelerator: Enabling random precision switch for winning both adversarial robustness and efficiency[C]//MICRO. Piscataway: IEEE, 2021: 225-237.

【责任编辑: 孙向荣】