FERMILAB-PUB-18-089-E

为提交给JINST而准备的

用于粒子物理学的FPGA中深度神经网络的快速推理

Javier Duartea, Song Hanb, Philip Harrisb, Sergo Jindariania, Edward Kreinarc, Benjamin Kreisa, Jennifer Ngadiubad, Maurizio Pierinid, Ryan Riveraa, Nhan Trana, Zhenbin Wue

[a] *费米国家加速器实验室，美国伊利诺伊州巴达维亚市，60510。*

[b] *麻省理工学院，剑桥，马萨诸塞州02139，美国*

[c] *HawkEye360, Herndon, VA 20170, USA*

[d] *CERN, CH-1211 Geneva 23, Switzerland*

[e] *伊利诺伊大学芝加哥分校，芝加哥，伊利诺伊州60607，美国*

*电子邮件：*hls4ml.help@gmail.com

摘要：大型强子对撞机（LHC）的最新成果表明，通过改进实时事件处理技术，物理学能力得到了增强。机器学习方法无处不在，并被证明在LHC物理学以及整个粒子物理学中非常强大。然而，在低延迟、低功耗的FPGA（现场可编程门阵列）硬件中使用此类技术的探索才刚刚开始。基于FPGA的触发器和数据采集系统具有极低的、亚微秒级的延迟要求，这是粒子物理学所特有的。我们提出了一个在FPGA中进行神经网络推理的案例研究，重点是喷气机亚结构的分类器，这将使我们能够在许多其他物理学场景中搜索新的暗部门粒子和希格斯玻色子的新测量。虽然我们关注的是一个具体的例子，但其教训是深远的。这项工作的配套编译器包是基于高级合成（HLS）开发的，称为hls4ml，用于在FPGA中建立机器学习模型。HLS的使用增加了广大用户群体的可及性，并允许大幅度减少固件的开发时间。我们绘制了FPGA资源使用和延迟与神经网络超参数的对比图，以确定粒子物理学中的问题，这些问题将受益于用FPGA进行神经网络推理。对于我们的例子喷气机子结构模型，我们在现代FPGA的可用资源范围内很好地适应了100 ns的延迟。

内容

1 简介

在过去的几十年里，随着物理学家对粒子物理现象的深入了解，粒子探测器被做得更大、更细，并且能够以越来越快的速度处理数据。这导致了数据量的急剧增加，需要对其进行实时有效的分析，以重建和过滤感兴趣的事件。部署在数据处理最后阶段的机器学习（ML）方法已被证明在整个粒子物理学的许多不同任务中极为有效。到目前为止，由于实施的复杂性和FPGA的资源需求，它们在基于现场可编程门阵列（FPGA）的实时选择硬件中的使用是有限的。在这项研究中，我们探索了神经网络在FPGA中的实现，描绘了各种深度神经网络架构和超参数的资源使用和延迟，证明了深度学习技术在非常低延迟（亚微秒）的FPGA应用中的可行性。

欧洲核子研究中心的大型强子对撞机（LHC）就是一个完美的例子。LHC是世界上能量最高的粒子加速器，以有史以来最高的数据速率运行。它的目标是了解自然界的基本规律和宇宙的构成要素。在2012年结束的第一次数据采集中，突破性的亮点是发现了希格斯玻色子[1，2]。目前的数据采集活动致力于全面描述希格斯玻色子的特性，并寻找粒子物理学标准模型之外的物理现象，包括寻找暗物质。

由于在大型强子对撞机上质子束碰撞的频率极高，CMS[3]和ATLAS[4]这两个多用途实验的数据率达到了每秒数百兆字节的水平。在如此高的数据率下，碰撞事件的实时和离线处理都面临着挑战。实时处理的任务是过滤事件，将数据率降低到可管理的水平，用于离线处理，称为触发。它通常在多个阶段进行[5，6]；CMS探测器中使用了两个阶段。由于极端的输入数据率和数据缓冲区的大小，数据处理的第一阶段，即第一级（L1），通常使用带有ASIC或越来越多的FPGA的定制硬件来处理初始数据率，使用具有数百纳秒和微秒的延迟的流水线算法。触发的第二阶段，高级别触发（HLT），使用商业CPU在软件中处理过滤后的数据，其延迟更长，总共有数百毫秒的时间尺度。

机器学习方法，最近是深度学习，在LHC的事件处理中有着广泛的应用[7-13]，从较低层次的能量簇校准和回归到高层次的物理对象分类，如带有子结构信息的射流标记，以及物理学分析。触发器中更复杂的ML算法将使大型强子对撞机实验保留潜在的新物理学特征，如与希格斯、暗物质和隐蔽部门有关的特征[14]，否则会丢失。这可以通过整体上更高性能的触发算法或快速数据分析技术来实现，如数据侦察或触发水平分析[15-17]。在这项研究中，我们通过绘制各种网络架构和超参数的资源使用情况和延迟，探索在FPGA中实现神经网络推理。目的是了解在不同的LHC事件处理应用的资源和延迟限制下，可能在FPGA中实现的神经网络设计范围。作为一个案例研究，我们考虑使用ML方法对喷气子结构[18]进行分类，在触发器中采用这种方法时，能够搜索新的暗部门粒子和希格斯横向动量谱的重要测量。这个案例研究中的经验教训是广泛适用的。

这项工作的一个重要补充成果是配套的编译器hls4ml[1]，它将Keras[19]和PyTorch[20]等常见开源软件包中的ML模型转换为RTL（寄存器-传输级）抽象，用于使用高级合成（HLS）工具的FPGA，该工具在最近几年已经显示出相当的进步[21]。有许多类型的HLS，我们的特定包和本研究的结果是基于Vivado HLS 2017.2 [22]，尽管一般的技术可以用于其他FPGA的应用。

在高能物理学中，需要具有较长开发周期的工程支持，以将以物理学为动机的数据处理算法转化为固件。然而，工程是一种稀缺而宝贵的资源。hls4ml工具允许物理学家在没有丰富的Verilog/VHDL经验的情况下，快速建立ML算法的原型，以保证固件的可行性和物理性能，从而大大减少了算法开发周期的时间，同时保留了工程资源。我们专注于ATLAS和CMS实验的基于FPGA的触发器的任务，其算法延迟在微秒范围内，完全管道化以处理40MHz的LHC碰撞率。对于这项任务，由于严重的时间限制，使用CPU或GPU的解决方案是不可能的。

[1] 该项目可在https://hls-fpga-machine-learning.github.io/hls4ml。

这样的延迟是大型强子对撞机触发挑战所特有的，因此很少有通用的工具可以用于这种应用。尽管如此，hls4ml软件包是一个通用工具，旨在为粒子物理学和其他领域的广泛应用服务，从触发和数据采集任务（DAQ）到更长的延迟触发任务（毫秒）和CPU-FPGA协处理器硬件。

本文的其余部分组织如下。在第2节中，我们描述了在FPGA中实现神经网络用于触发和DAQ的基本概念。这包括一个为FPGA实现开发喷气式子结构ML算法的案例研究。在第3章中，我们详细介绍了各种神经网络结构和超参数的HLS综合，以及由此产生的在FPGA上的实现。最后，我们总结了我们的发现，详细说明了后续研究，并在第4节讨论了在物理学和其他领域的潜在的更广泛的应用。

1.1 相关工作

FPGA中的神经网络推理是一个快速发展且备受关注的领域。关于这个主题有大量的文献。然而，由于它涉及到粒子物理学的特殊任务，其中网络可以更小，但延迟限制要严重得多，这是该领域中第一个专门的通用研究。尽管如此，一些ML技术已经被部署在LHC的触发器中，包括用于μ介子动量测量的提升决策树（BDT）的首次实施[23]。在NIPS 2017会议上，提出了在FPGA上为粒子物理学部署卷积神经网络（CNN）的早期尝试[24]。

我们从其他工作中获得灵感，包括RFNoC神经网络库[25]，而hls4ml正是基于此。在FPGA上映射CNN的现有工具流程的概述见[26]。Snowflake[27]是一个可扩展和高效的CNN加速器，其模型由Torch[28]指定，并采用单一的计算架构（顺序IO），旨在以接近峰值的硬件利用率执行，目标是Xilinx系统级芯片（SoC）。Caffeine[29]是另一个CNN加速器，用于Caffe指定的模型，目标是支持SDAccel15环境和FPGA与主机之间的PCIe接口的Xilinx器件。 fpgaConvNet[30-33]将Caffe[34]或Torch格式指定的CNN转换为生成的Xilinx Vivado HLS代码，采用流式架构（并行IO）。FP-DNN（现场可编程深度神经网络）[35]是一个框架，它将TensorFlow[36]描述的DNN（CNN、LSTM-RNN[37]和Residual Nets）作为输入，并通过RTL-HLS混合模板在FPGA板上生成硬件实现。DNNWeaver[38]是一个开源的替代方案，它也支持以Caffe格式指定的DNN，并使用手工优化的Verilog模板自动生成加速器Verilog代码，具有高度的可移植性。

我们讨论的物理学问题是基于亚结构的喷气式标签，关于这个问题有丰富的深度学习应用文献[39-46]。在这种情况下，有人尝试使用CNN、RNN以及物理学启发的网络架构。喷气机已被表示为灰度图像、RGB图像、粒子序列或一组物理学启发的高级特征，正如我们在本研究中所做的那样。

2 用hls4ml构建神经网络

在这一节中，我们概述了使用HLS将一个给定的神经网络模型转化为FPGA实现。然后我们详细介绍了一个具体的喷气机子结构的案例研究，但同样的概念也适用于广泛的问题类别。在本节的最后，我们讨论了如何在性能、资源使用和延迟方面创建一个高效和最佳的神经网络实现。

2.1 hls4ml概念

hls4ml软件包负责将训练好的神经网络（由模型的结构、权重和偏差指定）自动翻译成HLS代码。图1是一个典型工作流程的示意图。

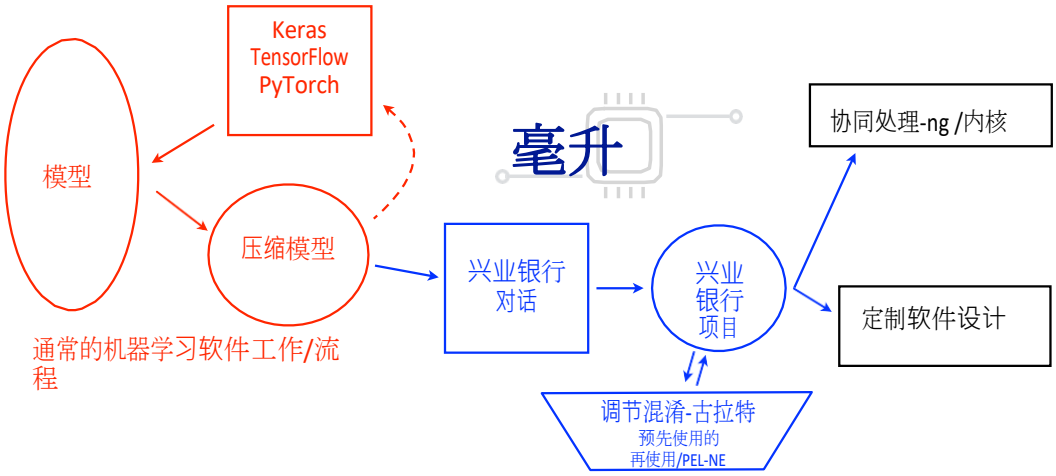

图1：使用hls4ml将模型翻译成FPGA实现的典型工作流程。

工作流程中的红色部分表示为特定任务设计神经网络所需的通常软件工作流程。这个通常的机器学习工作流程，使用Keras和PyTorch等工具，在确定最终模型之前，涉及训练步骤和可能的压缩步骤（更多讨论见下文第2.3节）。工作流程的蓝色部分是hls4ml的任务，它将一个模型翻译成一个HLS项目，可以被合成和实现，在FPGA上运行。

在高层次上，FPGA算法设计不同于CPU编程，因为独立的操作可以完全并行运行，使FPGA能够以相对较低的功率成本实现每秒数万亿次的操作，相对于CPU和GPU。然而，这些操作消耗了FPGA上的专用资源，并且在运行时不能动态地重新分配。创建一个最佳的FPGA实现的挑战是平衡FPGA的资源使用和实现目标算法的延迟和吞吐量目标。FPGA实现的关键指标包括。

1. **延迟**，即算法的一次迭代完成所需的总时间（通常以 "时钟 "为单位表示）。

2. **启动间隔**，即算法接受新输入前所需的时钟周期数。启动间隔（通常表示为 "II"）与推理率或吞吐量成反比；启动间隔为2时，吞吐量为启动间隔为1时的一半。

3. **资源使用情况**，以下列FPGA资源类别表示：板载FPGA存储器（BRAM）、数字信号处理（算术）块（DSP），以及寄存器和可编程逻辑（触发器，或FF，以及查找表，或LUT）。

hls4ml工具有许多可配置的参数，可以帮助用户探索和定制其应用的延迟、启动间隔和资源使用权衡的空间。因为每个应用都是不同的，hls4ml软件包的目标是使用户能够通过自动神经网络转换和FPGA设计迭代来进行这种优化。在实践中，对神经网络进行hls4ml翻译所需的时间比为FPGA设计一个特定的神经网络结构要短得多（几分钟到几小时），并且可以用来快速建立机器学习算法的原型，而不需要专门的工程支持来实现FPGA。对于物理学家来说，这使得为触发器或DAQ设计物理算法变得更加容易和有效，因此有可能大大减少 *"物理学时间"*。

我们首先介绍一些关于深度全连接神经网络推理的术语和概念。考虑图2中的网络，有M层，其中每层m有N_m个神经元。输入层有N_1个输入神经元，输出层有N_M个输出神经元。每层的神经元输出值的向量用$\boldsymbol{x}_m$表示。对于m^{th}全连接层（$m > 1$）。

$$\boldsymbol{x}_m = g_m\left(\boldsymbol{W}_{m,m-1}\,\boldsymbol{x}_{m-1} + \boldsymbol{b}_m\right)。\tag{2.1}$$

其中，$\boldsymbol{W}_{m,m-1}$是第$m-1$层和*第*m层之间的权重矩阵，$\boldsymbol{b}_m$是偏置值，g_m是第m层的激活函数。矩阵$\boldsymbol{W}_{m,m-1}$的大小为$N_m \times N_{m-1}$，因此计算*第*m层的神经元值所需的乘法数也隐含为$N_m \times N_{m-1}$。

在hls4ml中，每个层$\boldsymbol{x}_m$的计算是独立和顺序进行的。推理是流水线式的，在其启动间隔后接受一组新的输入，如上所述。推理一个给定的神经网络所需的乘法总次数为。

$$\text{牛的乘法} = \sum_{m=1}^{M} N_{m-1} \times N_m。\tag{2.2}$$

非琐碎的激活函数，如sigmoid、softmax和双曲正切，是针对一系列输入值预先计算的，并存储在BRAM中。ReLU激活函数是用可编程逻辑实现的。神经网络超参数对延迟、吞吐量和资源使用的影响告知任何特定应用的最佳网络实现。

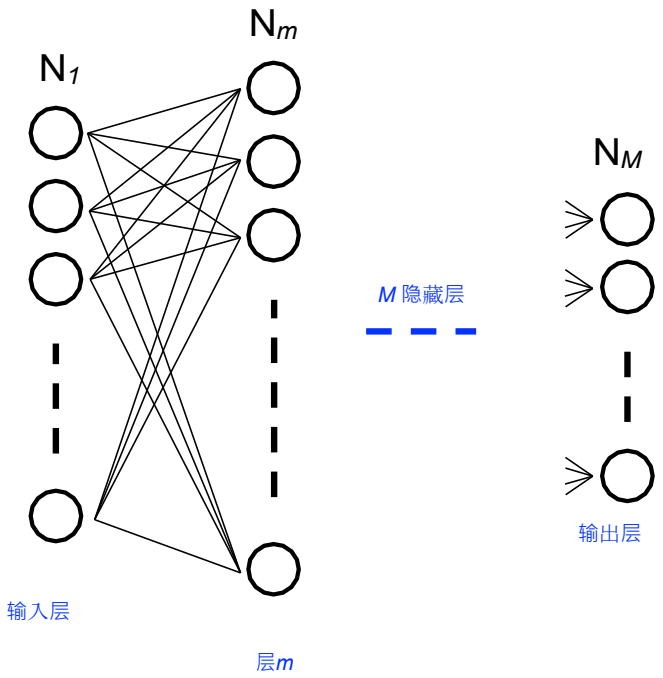

图2：一个深度全连接神经网络的漫画，说明了文中使用的描述惯例

2.2 案例研究：喷气式下层结构

喷流是由夸克q和胶子的衰变和强子化所产生的粒子的对撞喷流。在大型强子对撞机上，由于碰撞能量很高，在重标准模型粒子的衰变中产生的重叠夸克引发的喷流出现了特别有趣的特征。例如，W和Z玻色子在67%-70%的时间里衰变为两个夸克（$qq\bar{}$），而希格斯玻色子被预测在大约58%的时间里衰变为两个b-夸克（$bb\bar{}$）。顶夸克衰变为两个轻夸克和一个b-夸克（$qq\bar{}b$）。射流子结构[18, 47]的任务是将这些射流的各种辐射轮廓从主要由夸克（u, d, c, s, b）和胶子引发的射流组成的背景中区分出来。射流子结构的工具已经被用来从具有比信号大几百倍的生产率的背景中区分出有趣的射流特征[48]。

大型强子对撞机的射流亚结构一直是机器学习技术的一个特别活跃的领域，因为射流包含O(100)个粒子，其特性和相关性可以被用来识别物理信号。相空间的高维度和高度相关的性质使得这项任务成为机器学习技术的一个有趣的测试平台。有许多研究在实验和理论中探索这种可能性[18, 39-47, 49-51]。出于这个原因，我们选择使用喷气式亚结构任务来测试我们的FPGA研究。

我们在图3中举了两个例子，触发器中的射流子结构技术可以发挥重要作用：低质量隐性强子共振[52]和胶子聚变中产生的助推希格斯[53]。这两个过程都被背景所淹没，目前的触发策略会

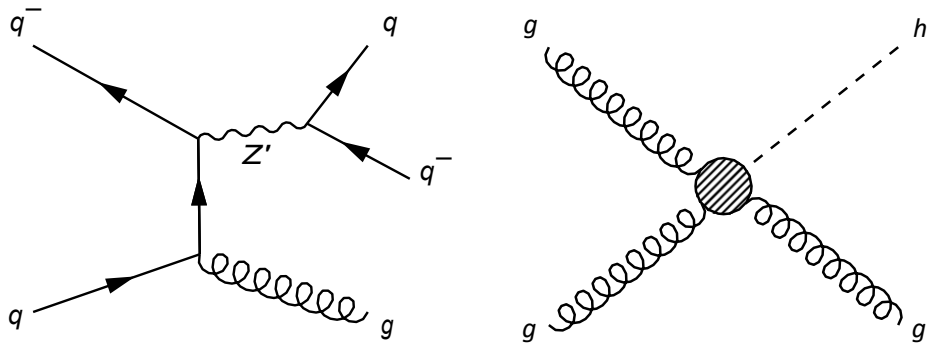

图3：有趣的物理学特征的费曼图示例，这些物理学特征将受益于硬件触发器中的喷射子结构算法。

只对射流的能量进行选择。通过在硬件触发器中引入射流子结构技术，我们可以进一步大大减少背景，并在未来大大保留更多的这类信号。还有许多其他的物理特征可以从触发器中的射流亚结构中受益。在这个案例研究中，我们重点关注将射流分类为夸克（q）、胶子（g）、W玻色子（W）、Z玻色子（Z）或顶夸克（t）射流[2]的任务。

输入的生成和特征

事件是在$\sqrt{s}$ = 13 TeV下产生的，以便与LHC的性能进行比较。首先使用MadGraph5_aMC_at_NLO[54]（2.3.1版）和NNPDF23LO1粒子分布函数（PDFs）[55]在前导阶上产生了粒子级（未施压的夸克）W^+W^-、ZZ、tt^-、qq^-和gg等事件。为了专注于一个rel-

在一个非常狭窄的运动范围内，在以1 TeV为中心的能量分布$\delta p_T / p_T = 0.01$的窗口中，产生了粒子和未衰变的规整玻色子的横动量。然后在Pythia8[56](8.212版)中用莫纳什2013年的调子[57]对这些质子级事件进行衰变和喷淋，包括基础事件的贡献。对于每个最终状态，产生200,000个事件。

为了建立一个完整的专家特征列表，我们通过FastJet 3.1.3和FastJet contrib 1.027软件包[58, 59]来实现各种喷流重组算法和子结构工具。作为一个基线，所有射流都使用反k_T 算法[60]进行聚类，其距离参数为$R = 0.8$。尽管质子级p_T 分布很窄，但由于来自质子雨的运动学反冲和能量在射流锥体内外的迁移，射流p_T 谱明显变宽。我们对重建的射流p_T 进行切割，以从分析中去除极端事件，否决那些在0.8 TeV < p_T < 1.6 TeV的窗口之外的p_T = 1 TeV bin。

[2] 本研究不包括希格斯玻色子，因为它的质量和子结构与W和Z玻色子相当相似，否则在没有轨道顶点信息的情况下很难区分，这种情况在目前基于FPGA的触发器中很常见。

观察者
mmMDT $N\beta=1'$ $M\beta=1'$ $C\beta=0'$ $C\beta=1'$ $D\beta=1'$ $D^{(\alpha,\beta)}$=(1,1), (1,2) z log 多重性

表1：分析中使用的观察指标的摘要。

射流子结构界已经开发了各种各样的观测指标，以根据射流辐射模式的结构来确定射流的起源。在表1中，我们列出了本研究中使用的所有观测指标[61-64]。对这些变量的简要描述见文献[65]。[65].这些都是作为专家级的输入到一个接近最优的神经网络分类器中3。

基准网络和浮点性能

我们为q、g、W、Z和t的分类任务训练一个神经网络。数据被随机分割成训练（60%）、验证（20%）和测试（20%）数据集。输入的特征通过去除平均数和缩放为单位方差而被标准化。图4（左）所示的结构是一个具有三个隐藏层的全连接神经网络。隐蔽层的激活函数是ReLU[66]，而输出层的激活函数是一个softmax函数，为每个类别提供概率。1使用Adam算法[67]对分类交叉熵损失函数进行最小化，初始学习率为10-4，最小批次大小为1024，权重的正则化（第2.3节）。如果验证损失在10个epochs内未能改善，学习率将减半。训练是在AWS EC2 P2 GPU实例[68]上进行的，使用Keras。在研究特定设备上的最终FPGA实现时，我们还考虑了一个具有一个隐藏层的更简单的结构，见图4（右）。这将在第3.3节进一步描述。

神经网络分类器的性能显示在图5中。这个性能图的一般特征是典型的射流子结构分类任务。顶夸克喷气，由于其大质量和三棱柱的性质，与其他喷气类型有最好的分离。W和Z射流由于其质量和双管齐下的性质，性能相似，而夸克和胶子射流的分类是出了名的挑战[48]。鉴于这种多射流分类器的性能，我们探索如何使用hls4ml在FPGA中实现这样一个神经网络架构。

3 存在更复杂的方法，但本研究的目标不是要实现比现有算法更好的性能。相反，目标是研究几种有效的神经网络架构在FPGA中的实现。

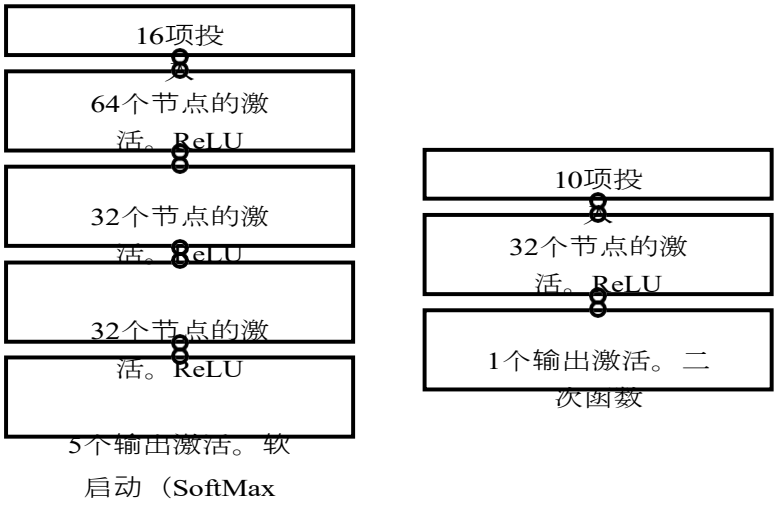

图4：用于喷气机子结构分类的两种神经网络结构。(左图）一个三层隐藏的模型，我们用来对五类射流（*q*、g、*W*、*Z*和*t*）进行分类。(右图）一个用于识别顶夸克的单隐层模型，为第3.3节所述的FPGA实现而简化。

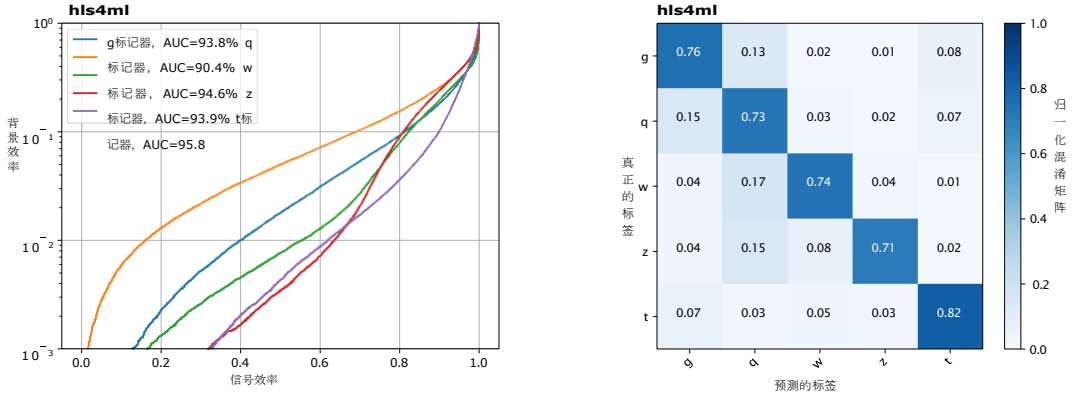

图5：深度神经网络分类器的性能：（左）夸克、胶子、*W*玻色子、*Z*玻色子和顶夸克射流识别的信号效率与错误识别率。错误识别率是基于其他非信号射流类型的平等混合。(右图)五个类别的相应归一化混淆矩阵。

2.3 高效的网络设计

在第2.1节中，我们介绍了将深度神经网络转化为FPGA实现的一般描述，并在第2.2节中介绍了用于喷气机亚结构分类任务的具体网络设计。现在，我们将重点放在调整网络推理的方式上，以使用FPGA

有效的资源，并满足延迟和流水线约束。

神经网络推理可以通过以下技术变得高效：压缩、量化和并行化。我们简要地总结一下这些想法。

- **压缩**：神经网络的突触和神经元可能是多余的；压缩试图减少突触或神经元的数量，从而有效地减少$N_{multpliers}$而不遭受任何性能损失。
- **量化**：在网络推理中通常不需要32位浮点计算来实现最佳性能；量化可以降低神经网络中的计算精度（权重、偏差等）而不损失性能。
- **并行化**：人们可以调整给定层计算所需的乘法的并行化程度；在一个极端，所有乘法可以使用最大数量的乘法器同时进行，而在另一个极端，人们可以只使用一个乘法器并按顺序进行乘法；在这些极端之间，用户可以优化算法吞吐量与资源使用。

在以下几个小节中，我们将更详细地描述这些优化的实现和效果。

我们没有讨论的一个重要话题是，在输入到神经网络之前如何计算输入特征，因为这取决于具体的应用。例如，在我们考虑的喷气机子结构案例研究中，专家特征的预计算可能相当耗费时间和资源。然而，在其他情况下，神经网络可能会接受原始检测器的输入，不需要进行什么预处理。在更现实的情况下，对神经网络输入的计算时间的考虑是一个重要的考虑。此外，考虑输入的精度和范围也很重要；位移或将输入转化为适当的范围可能对算法的性能很重要。

压缩

网络压缩是一种广泛的技术，可以减少深度神经网络的大小、能量消耗和过度训练[69]。有几种方法已经成功部署了压缩网络，包括[70]。

- **参数修剪**：根据特定的排名，有选择地删除权重[69, 71, 72]。
- **低等级因式分解**：使用矩阵/张量分解来估计信息参数[73-77]。
- **转移/紧凑卷积滤波器**：特殊结构的卷积滤波器以节省参数[78]，以及
- **知识提炼**：用大型网络的提炼知识来训练一个紧凑的网络[79]。

我们的方法是迭代参数修剪和重新训练的简化版[69, 80]，具有
L_1 正则化，其中损失函数L被增加了一个额外的惩罚项。

$$L_\lambda(\boldsymbol{w}) = L(\boldsymbol{w}) + \lambda \mathrm{l}/\mathrm{w}\,\mathrm{l}/_1 。 \tag{2.3}$$

众所周知，L_1 正则化可以产生稀疏的模型，提供内置的特征选择[81]，并且是许多机器学习工作流程中的一个现成选项。原则上，用$0 \le p < 1$的L_p 正则化训练[72]可以提高模型的稀疏度和性能，但这些正则化器并不总是容易实现。虽然我们采取了这种简化的方法，但我们注意到，文献中还有其他更复杂的压缩方法，可能会产生更好的结果。

我们用L_1 正则化训练模型，λ=10-4。然后，我们根据权重的绝对值相对于特定层中权重的最大绝对值进行排序。用
L_1 正则化，我们看到两个独立的权重子群，一个在较小的值，一个在较大的值。低于某个百分点的权重，对应于较小值的子群体，会被删除。接下来，我们再次用L_1 正则化来重新训练模型，同时约束之前修剪的权重保持为零。我们在这个过程的七次迭代后停止，此时，修剪后的权重子群体的总和是原始权重群体总和的3%，模型被压缩了70%（在4389个原始权重和偏差中修剪了3051个权重）。图6说明了这个过程。图6的左上方显示了压缩前的权重分布。从左上角到右下角，箭头表示剪枝和重新训练程序的后续步骤，并显示了所产生的权重分布。最后，在右下方，我们展示了压缩后的权重的最终分布。与原始网络相比，我们观察到修剪后的网络性能没有明显变化。

量化

量化[69, 82-85]甚至二值化[86-89]的神经网络已经被详细研究过了，这是一种通过减少表示每个权重所需的比特数来压缩神经网络的额外方式。FPGA在选择数据类型和精度方面提供了相当大的自由度。为了防止浪费FPGA资源和产生额外的延迟，这两者都是需要考虑的。在hls4ml中，我们使用定点算术，它比浮点算术使用更少的资源和延迟。

每层的输入、权重、偏置、总和和输出（见公式2.1）都表示为定点数字。对于每一个，二进制点以上和以下的位数都可以根据使用情况进行配置。据广泛观察，可以在不造成性能损失的情况下大幅降低精度[85]，但这必须谨慎进行。在图7中，我们显示了在第2.3节中描述的压缩后的权重绝对值的分布。在这种情况下，为了避免权重的下溢/溢出，至少应该在二进制点上方分配三个比特--两个用于包络最大的绝对值，一个用于符号。神经元值，x_m ，以及用于计算它们的FPGA中的中间信号可能需要更多的比特来避免下溢/溢出。我们通过扫描物理学性能作为比特精度的函数来确定在二进制点*以下*分配的比特数。

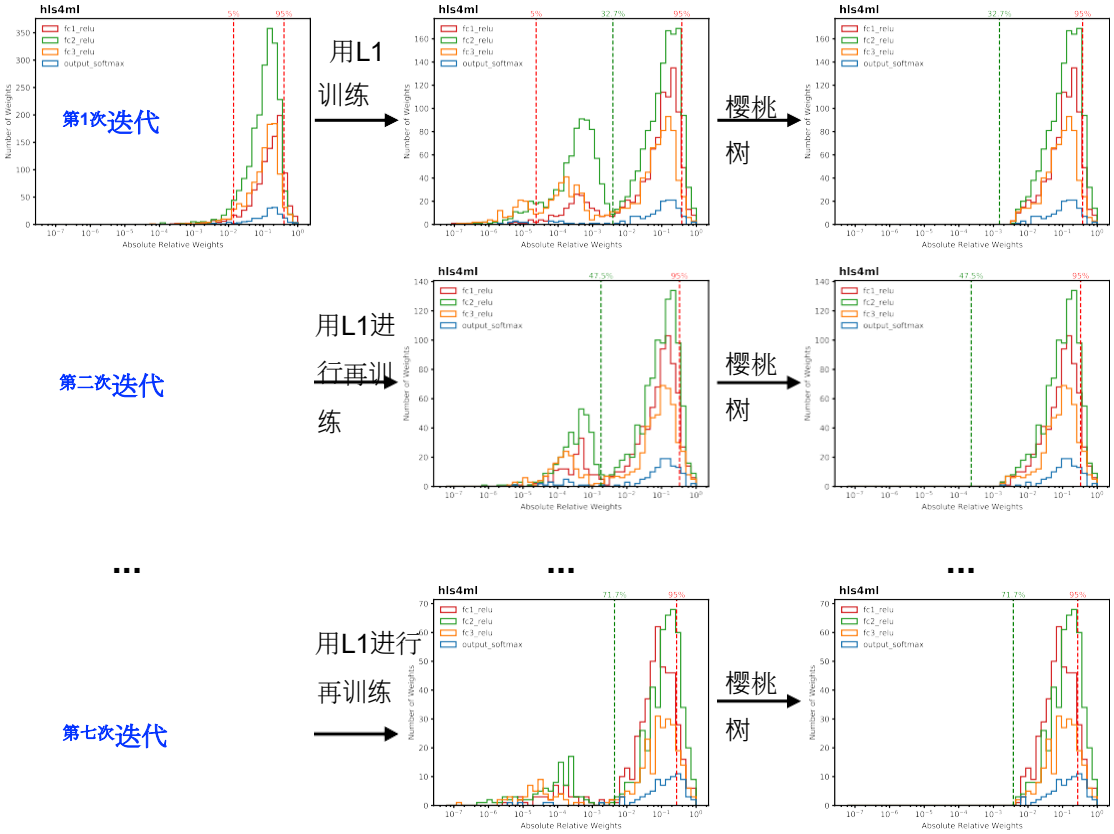

图6：用L_1 正则化程序进行的迭代参数修剪和再训练的说明。权重的绝对值相对于权重的最大绝对值的分布在修剪和再训练程序的每一步之后显示。左上方显示的是压缩前的分布，右下方显示的是压缩后的分布。

降低精度可以节省用于信号路由的资源，以及用于数学运算的资源和延迟。对于许多应用来说，限制FPGA资源的是DSP的数量，这些DSP主要用于乘法运算。每个乘法器使用的DSP数量取决于被乘数字的精度，并且可以突然改变。例如，一个赛灵思DSP48E1块[90]可以将一个25位数与一个18位数相乘，但要将一个25位数与一个19位数相乘则需要两个。同样，乘法器的延迟随着精度的提高而增加，尽管它们可以保持流水线。关于计算精度的影响的详细探讨将在第3节中介绍。

如第2.1节所述，非三段式激活函数是针对一定范围的输入值预先计算的，并存储在BRAMs中。这个范围内的分档和输出的位宽在hls4ml中是可以配置的。最后，我们注意到还有一些方法可以通过量化来进一步压缩网络结构，但本文没有对这些方法进行探讨[82, 88]。特别是，重新训练

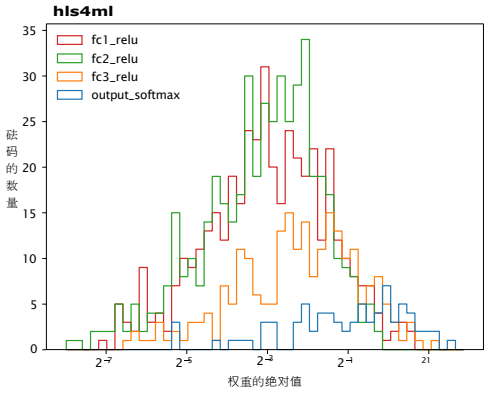

图7：压缩后的权重绝对值的分布。

在训练中采用量化精度的网络可以在权重精度明显较小的情况下获得同等的性能[91]。我们把对这些方法的调查留给进一步的工作。

并行化

延迟、吞吐量和FPGA资源使用之间的权衡是由推理计算的并行化决定的。在hls4ml中，这是用一个乘法器 "重用系数 "来配置的，该系数设定了在计算一个层的神经元值时使用乘法器的次数。如果重用系数为1，计算是完全并行的。当重用系数为R时，一次完成1/R的计算，乘法器减少1/R的系数。这在图8中得到了说明。

FPGA的乘法器是流水线式的；因此，一层计算的延迟，L_m，是近似的。
最近

$$l_m = l_{\text{mult}} + (r - 1) \times \text{ii}_{\text{mult}} + l_{\text{activ}} \text{。} \tag{2.4}$$

其中L_{mult} 是乘法器的延迟，II_{mult} 是乘法器的启动时间间隔，L_{activ} 是激活函数计算的延迟。方程2.4是近似的，因为在某些情况下，信号路由可能会产生额外的延迟，例如在对神经元值有贡献的乘法结果的增加。

正如第2.1节中所讨论的，我们独立地、按顺序地实现每一层的计算。在前一层的计算完成之前，不能启动某一层的计算。因此，总延迟等于每一层的延迟之和加上连接各层所需的延迟。每单位时间内完成的推理数量与重用系数成反比。

3 业绩和实施

在这一节中，我们对第2.2节中描述的喷气式子结构神经网络的HLS翻译和优化结果进行量化，作为前面描述的三个基本原则的功能。

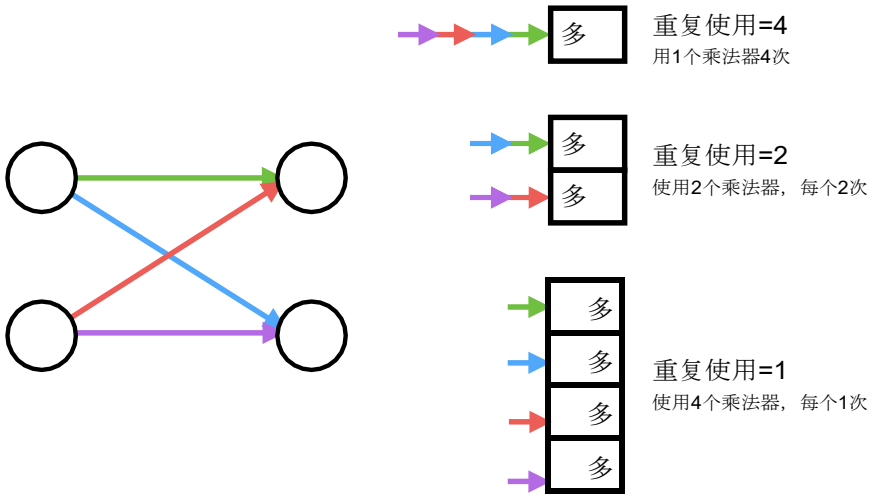

图8：不同重用系数值下的乘法器资源使用说明。左图显示了两个神经元对被4个连接所连接的情况，意味着要进行4次乘法。右图显示了如何执行这些乘法，从完全串行（顶部）到完全并行化（底部）。

节：压缩、量化和并行化。首先，我们在第3.1节中讨论了在固件中实现的神经网络的分类性能。然后在第3.2节中，我们从FPGA资源使用和延迟的角度对HLS综合进行量化。这两个指标的组合，即分类和固件性能，定义了如何为一个给定的应用在FPGA硬件中最佳地实现神经网络。最后，在第3.3节中，我们讨论了特定FPGA的实现，并将实际的资源使用情况与Vivado HLS的估计值进行比较，后者可以更快地获得。

3.1 分类性能

为了量化我们的五输出分类器的性能，我们使用了AUC指标，或接收操作特征（ROC）曲线下的面积。如图5所示，ROC曲线是由分类器输出上的顺序切割计算出的背景排斥与信号效率给出的，其中背景排斥是（1−背景效率）。我们将神经网络的全32位浮点推理所达到的AUC表示为预期AUC。我们用<X,Y>表示的定点精度来评估神经网络，其中Y是代表二进制点以上的有符号数字（即整数部分）的比特数，X是总比特数。我们进行两次扫描--一次是固定整数位数，一次是固定小数位数。结果如图9所示，左边是对整数位的扫描，右边是对小数位的扫描。

不损失分类能力的最佳性能对应于AUC/Expected AUC = 1。图9显示，通过定点计算和足够的比特数，预期AUC可以再现，而性能损失可以忽略不计。整数位的数量被选择为仅为

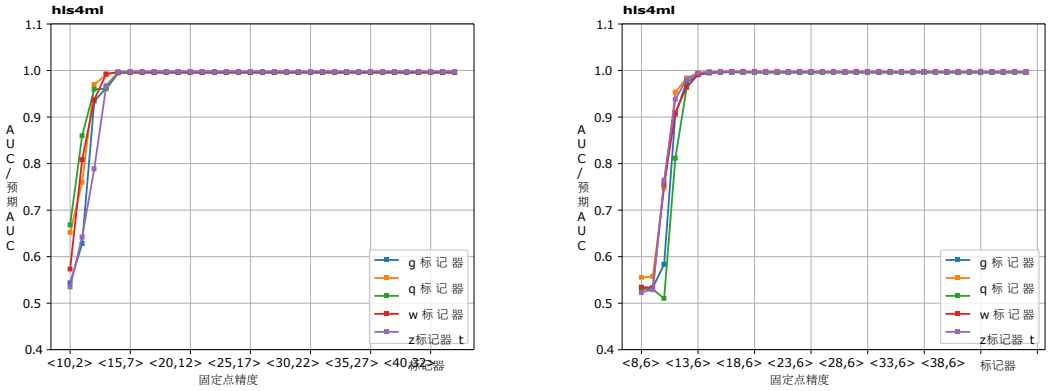

图9：全连接三层隐蔽网络的定点AUC和预期AUC与定点精度的比率。不损失分类能力的最佳性能对应于比率为1。（左）整数位的数量被扫描。(右)整数位的数量被固定为6，小数位的数量被扫描。各种颜色的线是不同喷气式子结构标记器（q,g,W,Z,t）的AUC性能。

以上的点，在这个点上不会发生欠流/溢出，AUC/预期AUC=1。有了这个整数位的数量，我们再扫描小数位的数量。总共约16位就能达到最佳性能。

我们进行类似的扫描，将压缩的三层隐藏模型AUC与未压缩的模型进行比较。如图10所示，与预期AUC的一致性发生在大致相同的精度上。

3.2 HLS中的延时和资源估算

我们现在探讨该模型所需的FPGA资源是如何被以下因素影响的

- **压缩**，三层隐藏式模型，70%的参数被修剪。
- **量化**、输入的精度、权重和偏置；对于这个特定的网络，根据我们在第3.1节的讨论，我们着重于定点精度<X,6>的扫描。我们在达到最佳性能的点以上进行扫描，以显示量化的好处和需要更高的精度时的资源使用。
- **并行化**，即一个给定的乘法器被用于一个层的计算的次数；使用一个乘法器一次是一个层可以被计算的最并行（和快速），也就是我们所说的*重用系数*为1。

以这些变量作为如何控制网络实施的把手，我们监测以下固件实施指标。

- **资源**。DSP、FF和LUT。

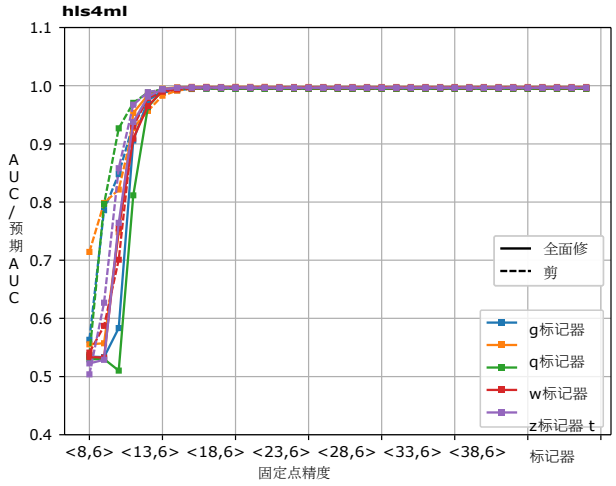

图10：固定点AUC和预期AUC与固定点精度的比率。实线代表完全连接的网络，而虚线表示压缩的网络。

- **延迟**：计算整个网络所需的时间。
- **启动间隔**：在接受一组新的输入之前的时间。

目前，我们没有探测块状RAM（BRAM）的使用情况，它只用于存储预计算的激活函数值。例如，从BRAM中存储和访问神经网络权重，会导致延迟超过LHC L1触发器第一阶段的要求。对于更长的延迟任务，例如HLT应用，hls4ml的能力可以扩展到允许在BRAM中存储权重。

下面介绍的结果是在Xilinx Kintex Ultrascale FPGA上合成的，部件号为xcku115-flvb2104-2i。时钟频率被固定在200MHz，这是典型的LHC触发器的第一阶段。结果与时钟频率可能会有变化，但在O(100 MHz)范围内，我们发现变化是可以忽略的。资源使用估计来自于Vivado HLS综合步骤，并且发现与实施相比，总体上是保守的，我们将在第3.3节讨论。虽然保守，但由于进行HLS资源估算所需的时间很短，因此对于快速制作不同的网络设计原型和得出有用的趋势非常有用。第3.3节讨论了我们的结果对Vivado HLS版本的依赖性，具体的FPGA，以及在FPGA中的最终实现。

具有压缩功能的资源

我们首先探索压缩对神经网络所需的FPGA资源的影响。因为压缩通常是训练工作流程的一部分，我们把它与其他优化处理分开考虑。看一下图11中的DSP用量和算法延迟。

我们展示了我们的压缩和未压缩的神经网络模型之间的差异。在这两种情况下，我们认为网络都是最大限度的并行化（*重用系数*为1）。由于权重存储在可编程逻辑中，稀疏矩阵乘法被简单处理，零权重乘法被优化出网络FGPA实现。我们发现这是HLS的一个非常有吸引力的特点，尽管像[92]中描述的那些更复杂的压缩技术可能需要更多研究。

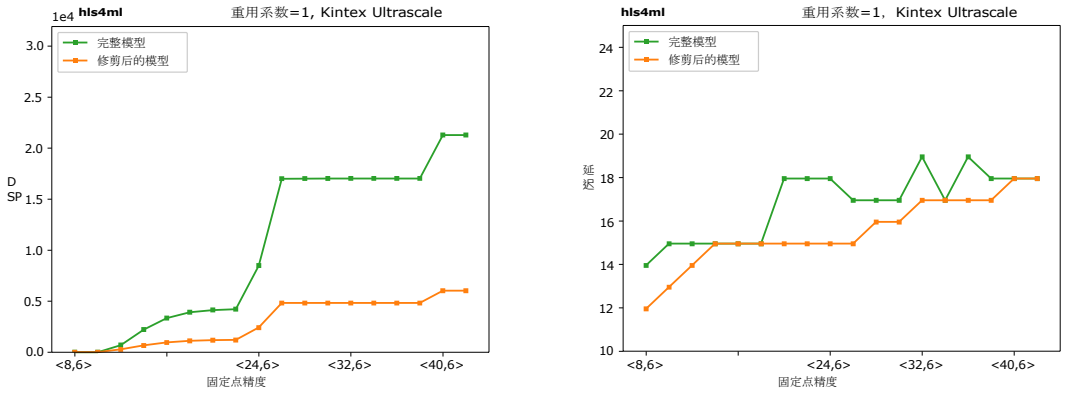

图11：压缩模型和未压缩模型的比较，DSP使用的重用系数为1（左），时钟频率为200MHz的延迟为时钟周期（右）。X轴是对模型的定点精度的扫描，展示了资源使用量是如何作为网络推理中计算精度的函数而变化的。

如图11（左）所示，与原始网络相比，压缩模型的DSP使用量大幅减少，与第2.3节所述的70%的压缩率成正比。此外，DSP的使用量随着定点精度的提高而增加。这种增加不是平滑变化的，因为它们取决于DSP的设计精度。在图11的右边，我们展示了在200MHz时钟频率下算法的延迟，以时钟周期为单位。由于网络仍然具有相同的结构，就隐藏层的数量而言，在压缩和未压缩的模型中，延迟是大致相同的。请注意，推断模型的总延迟约为15个时钟周期，转化为75纳秒，完全在LHC触发器第一阶段的延迟预算之内。

为了总结压缩和非压缩模型的HLS合成的结果，我们在表2中报告了一些重要的统计数据。我们注意到在保持相同的性能、延迟和启动间隔的情况下，资源有所减少。

压缩的三层隐藏式模型结果

我们现在考虑将我们的压缩三层隐藏式神经网络模型作为我们用例的基准模型，并对FPGA资源与网络精度和重用系数进行详细扫描。在图12和图13中，我们检查了DSP、FF和LUT的使用情况，作为网络精度的函数。

网络	未压缩的网络	压缩的网络
AUC / 预期AUC	99.68%	99.55%
参数	4389	1338
压缩系数	-	3.3×
DSP48E	3329	954
逻辑（LUT + FF）	263,234	88,797
延迟	75 ns	75 ns

表2：在Xilinx Kintex Ultrascale FPGA上合成的网络精度为定点<16,6>、全流水线、时钟频率为200MHz的未压缩和压缩的喷气式子结构标记模型的重要统计数据和HLS资源估算。

固定点计算，<X,6>。根据第3.1节的发现，我们通过扫描X来扫描小数位的数量，同时将整数位固定在*Y*=6，保证没有下溢/上溢。对于不同的重用系数值显示了不同的曲线。

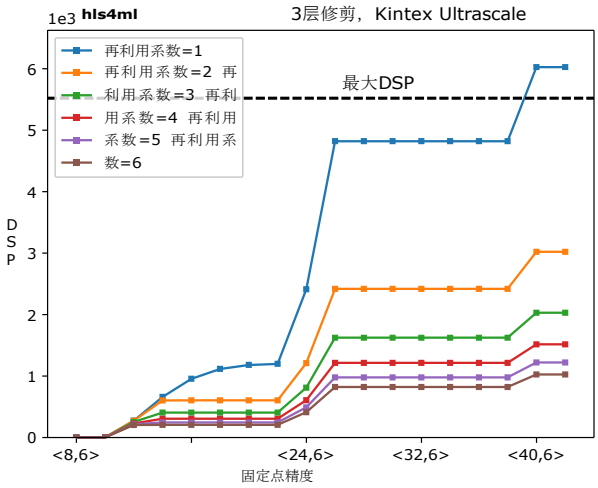

图12：压缩的三层隐藏模型中的DSP使用量与网络精度的关系。各条曲线说明了不同资源使用系数的资源使用情况。

在图12中，我们展示了重用系数是如何用来控制神经网络中使用乘法器的次数的。随着重用系数的增加，我们能够按重用系数的比例控制DSP的使用。DSP的资源使用量与网络精度的函数有阶梯关系。这对于所有的重用值都是一致的，并且来自于Xilinx FPGA DSP的精度。在图中，我们还指出了这个特定的Xilinx Kintex Ultrascale FPGA中可用的DSP的最大数量。在图13中，显示了LUT（左边）和FF（右边）的使用情况。对于LUT和FF来说，相对于FGPA的容量来说，资源使用量很小。

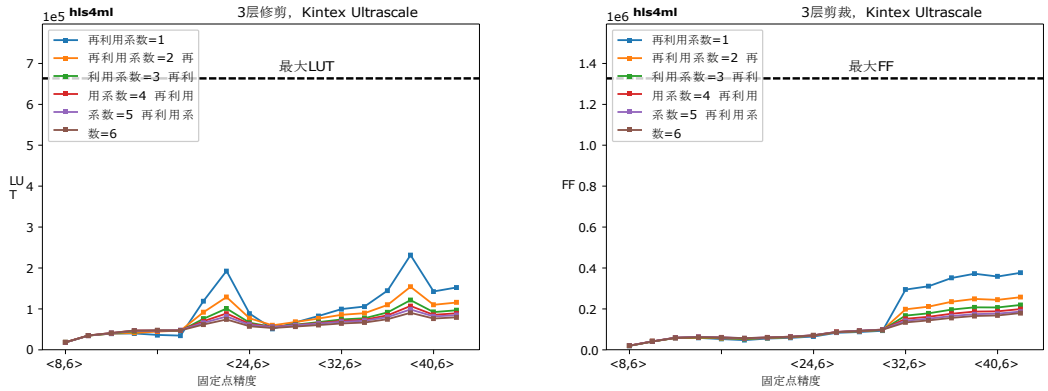

图13：在压缩的三层隐藏模型中，LUT和FF的使用是网络精度的一个函数。各条曲线说明了不同资源使用系数的资源使用情况。

与DSP的相比。此外，我们观察到在DSP精度限制下的FF使用峰值。我们发现，在进行实现时，它们被消除了（在第3.3节讨论）。我们注意到在图12和图13中的一个总体趋势，即当重用率>1时，重用率=1的情况会偏离其他情况的趋势。我们相信在重用率=1的情况下，HLS能够对一个给定的网络设计的单个乘法器做进一步的优化，而当重用率>1时，一个乘法器被赋予多个操作，它就没有这种优化自由。

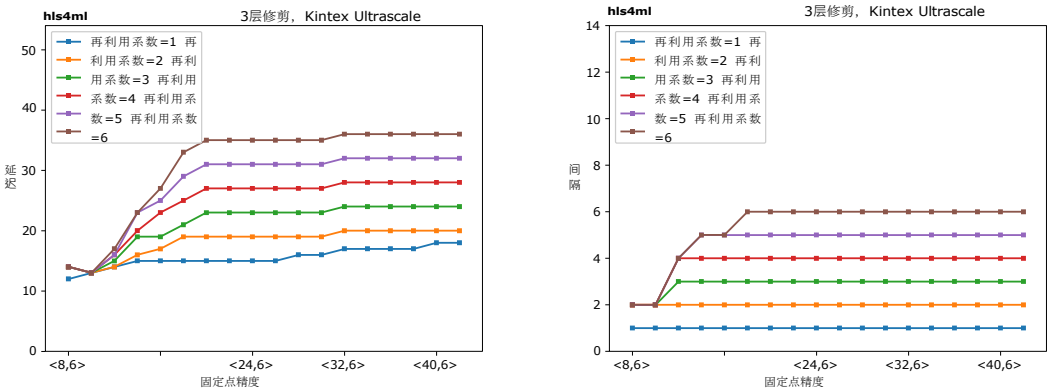

图14：压缩的三层隐藏模型中的延迟（左）和启动间隔（右）是网络精度的函数。各条曲线说明了不同资源使用系数的资源使用情况。延迟的单位是时钟频率为200MHz的时钟周期。

接下来，我们检查FPGA实现的与延迟和吞吐量有关的方面。在图14中，在左边（右边），我们显示了在一些不同的重用因素下算法的延迟（启动间隔）与精度的关系。网络推理的延时增加了4个时钟。

对应于必须计算的四层神经元值，每增加一个重用系数。这符合公式2.4的预期，即在一个给定的层计算中额外的重用乘法器会产生额外的延迟。根据设计，启动间隔和重用系数是匹配的，因为只有当一个给定的乘法器的所有乘法运算完成后，才能将一个新的输入引入算法中。在非常低的网络精度下，HLS合成的启动间隔比重用系数要小，因为乘法不再在DSP中实现，而是通过FF和LUT实现。

3.3 FPGA实施

为了对Vivado HLS资源估算和最终的 "放置和布线 "实现之间的差异有一个定性的了解，我们使用了一个 "裸 "固件设计，除了神经网络所需的资源外，使用的资源最少。这个 "裸 "实现包括一个简单的VHDL包装器，它将hls4ml固件块直接连接到FPGA的通用输入/输出引脚，以防止Vivado优化掉我们试图描述的逻辑。包括VHDL包装器在内，我们进行了固件实现，并将结果的资源使用量与Vivado HLS的估计值进行了比较。

在进行实现时，我们注意到，由于设计在实现步骤中没有满足时序约束，最初没有达到HLS的目标时钟周期，因此我们在最终的FPGA实现中增加了时钟周期以满足时序约束。所需增加的时钟周期随着NN的复杂性而变大；占用FPGA很大一部分的算法需要更长的时钟周期。对于32位精度的三层隐藏式神经网络，需要8ns的时钟周期来实现设计为5ns的HLS块。这种情况在所有的重用因子中都可以看到。克服这个问题的一个简单的办法是对HLS设计进行综合，使之比预定的时钟周期略小。我们还注意到，不同版本的Vivado HLS在满足时序约束方面有不同程度的成功。我们用Vivado 2016.4比2017.2更成功地满足HLS目标的时序约束。

图15显示了功率使用情况。对于所有的实现，一个明显的趋势是更大的网络精度的电力使用。正如预期的那样，随着重用系数的增加，吞吐量下降，功率使用也会下降。

我们将每一位输入/输出直接连接到一个独特的引脚上，假设所有的输入都在同一个时钟沿上传递。在这个 "裸 "实现中，我们没有使用高速收发器，而高速收发器会使输入/输出的带宽大大增加。由于引脚的数量有限，我们现在考虑一个不同的神经网络模型，输入较少。该模型如图4（右）所示，有十个输入神经元和一个输出神经元，中间有一个隐藏层。我们还测试了三层隐藏层的修剪网络，并在引脚数量足以实现的制度中发现类似的定量结论。在本小节的其余部分，我们介绍了在执行时使用8ns时钟的单隐藏层网络的结果。

图16显示了与HLS合成得到的DSP估计值相比，实现的设计的DSP使用情况。在所有的情况下，实现的设计中的DSP使用量都小于HLS的估计值，特别是，我们发现HLS的综合估计值和实现的设计在需要一个DSP（<24位）的乘法上非常一致。两者之间的偏差

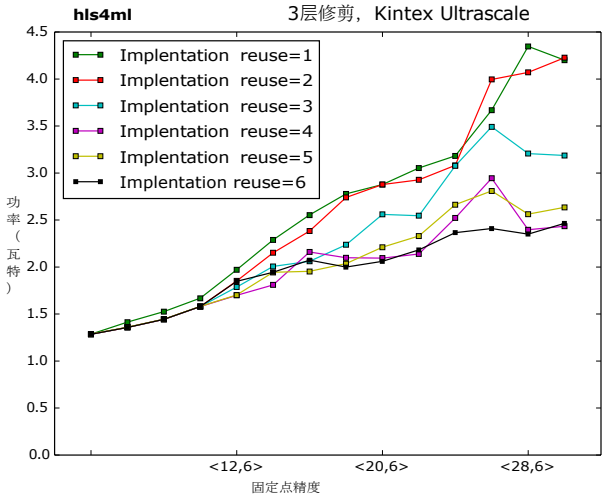

图15：基线三层隐藏模型的功率使用（W）与精度的关系

然而，HLS估算和Vivado实现在24位以上是很明显的，因为其他FF和LUT资源可以进一步用于优化DSP在最终实现中的使用。

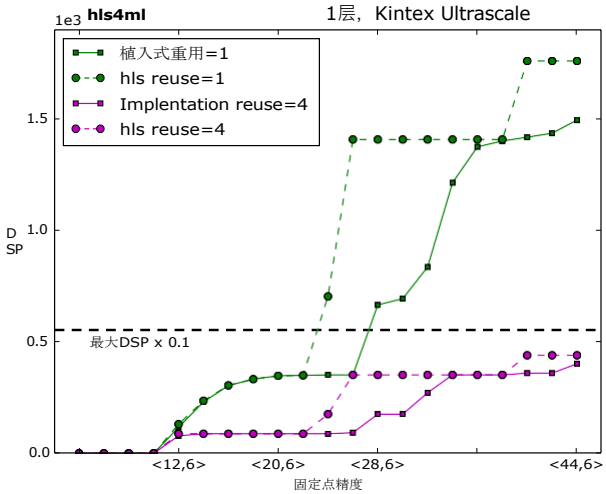

图16：在Xilinx Kintex Ultrascale FPGA上，单层隐藏式实现的DSP使用量与各种重用因素的精度的比较。

图17比较了HLS估计和实现之间的FF和LUT使用情况。在HLS的估计和实现的FF使用量之间存在着很大的差异。HLS的估计值通常是2-4倍，特别是在实现时使用了

超过32位。LUT的使用同样被HLS计算高估了，在植入步骤中，22位和38位的峰值被优化掉了。对于所有的点，不包括LUT的26位实现，HLS的估计比固件的实现更保守。

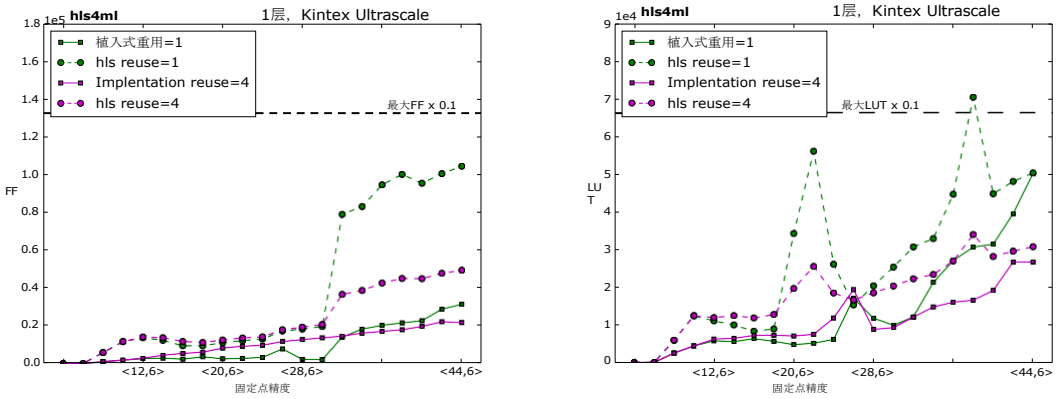

图17：Kintex Ultrascale处理器的FF性能（左）和LUT性能（右）的比较，作为1和4重用因子的精度的函数。

4 总结和展望

我们介绍了hls4ml，这是一个基于HLS的深度神经网络编译器，能够将完全连接的网络移植到FPGA上，从传统的训练框架如Keras和PyTorch训练。对于使用这个框架的第一个结果，我们专注于在LHC的实时事件重建和过滤的应用，在定制的电子产品中使用FPGA。这需要流水线式的网络推理，其延迟是1微秒的规模。对于这样的低延迟，网络必然具有较少的参数数量。在本文中，我们考虑了一个具体的案例研究，并训练了一个完全连接的神经网络，以识别喷气机是来自于轻夸克、胶子、*W*玻色子、*Z*玻色子或顶夸克。原始模型有4389个参数，应用网络压缩和降低精度，可以在Xilinx Kintex Ultrascale中用大约10%的可用DSP实现一个全连接的三层隐藏网络，结果随启动间隔而变化。在时钟频率为200MHz的情况下，推理的延迟大约为75-150ns。这非常符合LHC探测器（如ATLAS和CMS）允许的硬件触发重建预算。

HLS的可访问性和易配置性使得物理学家可以快速开发和优化针对FPGA硬件的机器学习算法。与传统的基于VHDL/Verilog的算法相比，这大大减少了固件的开发时间和物理学中稀缺的工程资源。我们讨论了诸如网络压缩、并行化和降低精度等通用技术，这些技术可以应用于设计高效的神经网络实现，为LHC及其他不同的应用量身定做。提出的结果使用hls4ml来扫描网络精度和并行化，以优化DSP和其他资源的使用。我们比较了

从HLS综合中估计的资源与实施时的资源使用情况的结果。HLS的资源估计是保守的，特别是对于FF和LUT。HLS对DSP使用的资源估计与实现的设计相当，尽管它在设计的DSP精度上可能是保守的。

虽然我们在全连接神经网络的背景下展示了hls4ml框架，但我们打算让hls4ml成为一个通用工具，用于转换物理学中常用的许多类型的神经网络架构。例如，我们设想将该框架扩展到卷积神经网络（CNN）和循环神经网络（RNN），如长短时记忆（LSTM）单元[37]。CNN通常用于基于图像的问题，包括热量计集群重建[39, 93]。LSTM用于处理对象的动态列表；对于喷气机的子结构标记，它们被用来处理属于单个喷气机的粒子属性列表。这两种网络结构的核心都是建立在现有的hls4ml框架之上，并且适用同样的高效网络设计原则。此外，hls4ml可以扩展到针对其他供应商的FPGA，例如使用Quartus HLS编译器的Intel FPGA。我们目前支持基于Keras和PyTorch的实现方式。

这条研究路线还有进一步有趣的延伸，即使用FPGA协处理器来处理机器学习，由Micrsoft Brainwave[94, 95]等人开创的。在CMS和ATLAS这样的实验中，第一层的重建被限制在微秒级的时间尺度。然而，第二层的重建，即高层触发器，仅限于重建和理解100毫秒时间尺度的事件。这第二层目前读取第一层的输出，其速率比原始碰撞速率减少了400倍。在这些时间尺度上，推断时间可能长达 10 毫秒。对于这样长的时间尺度，一个大的重用系数可以允许大型机器学习算法放在FPGA上。因此，FPGA可以作为一个协处理器加速器，大大减少执行复杂的、核心的LHC重建算法（如轨道重建）所需的时间。凭借比CPU快O(100)倍的推断能力[96]，FPGA可以作为一个低功耗、低成本的协处理器与CPU一起使用，可以用来大大加快高层触发器的速度，并可能提高其性能。在未来的工作中，我们期待着在CPU、GPU和FPGA硬件上对物理学应用的神经网络进行更详细的比较。

除了大型强子对撞机之外，其范围非常广泛。我们相信这个工具可以被用于许多不同的科学应用中。在核物理和粒子物理学中，越来越多的更强的光束和更高的速率的实验被开发出来，这些实验中的读出和处理往往需要对复杂的数据输入进行高速推理。

鸣谢

我们要感谢Evan Coleman、Marat Freytsis和Andreas Hinzmann在相关工作中协助制作数据集。我们感谢Jeffrey Berryhill、Doug Burger、Richard Cavanaugh、Eric Chung、Scott Hauck、Andrew Putnam、Ted Way以及Xilinx的成员提供的有用的对话。神经网络训练是在CERN和亚马逊AWS GPU资源上运行。亚马逊AWS GPU资源是通过费米实验室提供的，作为能源部 "现场工作建议 "的一部分，特别是。

我们要感谢Lothar Bauerdick和Burt Holzmann的支持。我们还感谢Xilinx和Ettus Research赞助2017年RFNoC和Vivado HLS挑战赛。J. D., B. K., S. J.。

R.R.和N.T.是由费米研究联盟有限公司根据美国能源部科学办公室高能物理办公室的合同号DE-AC02- 07CH11359支持的。

P.H.得到了麻省理工学院大学拨款的支持。M.P.和J.N.得到了欧洲研究理事会（ERC）在欧盟地平线2020研究和创新计划下的支持（资助协议*编号*772369）。Z. W.得到了美国国家科学基金会的支持，资助号为1606321和115164。

参考文献

[1] ATLAS合作，G. Aad等人，*在LHC的ATLAS探测器搜索标准模型希格斯玻色子中观察到一个新粒子*，*Phys. Lett.***B716** (2012) 1-29, [1207.7214].

[2] CMS合作，S. Chatrchyan等人，*在LHC的CMS实验中观测到一个质量为125GeV的新玻色子*，*Phys. Lett.***B716** (2012) 30-61, [1207.7235].

[3] CMS合作，S. Chatrchyan等人，*欧洲核子研究中心大型强子对撞机的CMS实验*，*JINST* **3**（2008）S08004。

[4] ATLAS合作，G. Aad等人，*欧洲核子研究中心大型强子对撞机的ATLAS实验*，*JINST* **3** (2008) S08003.

[5] ATLAS合作，C. Bernius，*ATLAS触发器算法升级和Run-2的性能*，2017。1709.09427.

[6] CMS合作，M. Tosi, *The CMS trigger in Run 2*, *PoS* **EPS-HEP2017** (2017) 523.

[7] CMS合作，A.M.Sirunyan等，在$\sqrt{s}$=*13TeV的质子-质子对撞中，二光子衰变通道的希格斯玻色子特性测量*，1804.02716。

[8] ATLAS合作，M. Aaboud等人，*在H的希格斯玻色子质量的测量。* $\rightarrow ZZ^* \rightarrow 4t$ *使用ATLAS探测器在*$\sqrt{s}$ = 13 *TeV pp对撞下的*$H \rightarrow \gamma\gamma$*通道*，1806.00242。

[9] CMS合作，A. M. Sirunyan等人，*在13 TeV的pp对撞中用CMS探测器识别重味喷流*，*JINST* **13**（2018）P05011，[1712.07158]。

[10] ATLAS合作，M. Aaboud等，*用ATLAS探测器在*$\sqrt{s}$ = 13 *TeV的*$t\bar{t}$*事件测量b-jet标记效率*，1805.01845。

[11] M.Lassnig, W. Toler, R. Vamosi, J. Bogado和A. Collaboration, *Atlas分布式数据管理中网络度量的机器学习*, *Journal of Physics:会议系列***898**（2017）062009。

[12] CMS合作，A. M. Sirunyan等人，*观察*ttH*的产生*，*Phys. Rev. Lett.***120** (2018) 231801, [1804.02610].

[13] ATLAS合作，M. Aaboud等人，*在LHC用ATLAS探测器观测希格斯玻色子产生与顶夸克对的关系*，1806.00425。

[14] Y.Gershtein, *CMS Hardware Track Trigger:在HL-LHC进行长寿命粒子搜索的新机会*，*Phys. Rev.* **D96** (2017) 035027, [1705.04321].

[15] CMS合作，V. Khachatryan等人，在$\sqrt{s}$ = *8的二喷子最终状态中搜索窄谐振* $\bar{(}$ *TeV的数据侦察的新型CMS技术*，*Phys. Rev. Lett.***117** (2016) 031802, [1604.08907].

[16] ATLAS合作，M. Aaboud等，*在sqrt(s)=13 TeV的pp对撞中，用ATLAS探测器的触发级喷气机搜索低质量的二喷共振*，1804.03496。

[17] CMS合作，A. M. Sirunyan等人，*在质子-质子对撞中寻找喷气式共振，时间为$\sqrt{s}$ ==。13 TeV以及对暗物质和其他模型的约束*，*Phys. Lett.***B769** (2017) 520-542, [1611.03568].

[18] A.J. Larkoski, I. Moult和B. Nachman, *大型强子对撞机的喷射结构。A Review of Recent Advances in Theory and Machine Learning*, 1709.04464.

[19] F.Chollet等人，"Keras"。https://keras.io， 2015。

[20] A.Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito et al., *Automatic differentiation in pytorch*, .

[21] G. Inggs, S. Fleming, D. Thomas and W. Luk, *Is high level synthesis ready for business? a computational finance case study*, in *2014 International Conference on Field-Programmable Technology (FPT)*, pp.12-19, December, 2014.DOI.

[22] D.O'Loughlin, A. Coffey, F. Callaly, D. Lyons and F. Morgan, *Xilinx vivado high level synthesis:案例研究，在2014年爱尔兰信号与系统会议和2014年中国-爱尔兰信息与通信技术国际会议（ISSC 2014/CIICT 2014）上。第25届IET年。*
pp.352-356, 2014.

[23] CMS合作项目，D. E. Acosta, A. W. Brinkerhoff, E. L. Busch, A. M. Carnes, I. -K.Furic, S. Gleyzer et al., *Boosted Decision Trees in the Level-1 Muon Endcap Trigger at CMS*, Tech.CMS-CR-2017-357, CERN, Geneva, Oct, 2017.

[24] T.Boser, P. Calafiura和I. Johnson, "卷积神经网络用于FPGA上的轨道重建"。https://docs.google.com/presentation/d/1mTqsm5TronnB8MFD6yyq3CSPCV4bfck_aQ-ericM9cQ/edit#slide=id.p3, 2017。

[25] E.Kreinar, *Rfnoc neural network library using vivado hls*, *Proceedings of the GNU Radio Conference* **2** (2017) 7.

[26] S.I. Venieris, A. Kouris and C.-S. Bouganis.Bouganis, *FPGA上映射卷积神经网络的工具流程。A Survey and Future Directions*, *ArXiv e-prints* (Mar., 2018) , [1803.05900].

[27] V.Gokhale, A. Zaidy, A. X. M. Chang and E. Culurciello, *Snowflake:深度卷积神经网络的模型不可知加速器*, *CoRR* **abs/1708.02579** (2017) , [1708.02579].

[28] R.Collobert, K. Kavukcuoglu and C. Farabet, *Torch7: A matlab-like environment for machine learning*, in *BigLearn, NIPS Workshop*, 2011.

[29] C.Zhang, Z. Fang, P. Zhou, P. Pan and J. Cong, *Caffeine:Towards uniformed representation and acceleration for deep convolutional neural networks*, in *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp.1-8, Nov, 2016.DOI.

[30] S.I. Venieris and C.-S.Bouganis, *fpgaConvNet:A Toolflow for Mapping Diverse Convolutional Neural Networks on Embedded FPGAs*, in *NIPS 2017 Workshop on Machine Learning on the Phone and other Consumer Devices*, 2017.

[31] S.I. Venieris and C.-S.Bouganis, *fpgaConvNet:卷积神经网络在FPGA上的自动映射（仅有摘要），在2017年ACM/SIGDA国际研讨会的论文集上。*
现场可编程门阵列，第291-292页，ACM，2017。DOI.

[32] S.I. Venieris and C. S. Bouganis, *Latency-Driven Design for FPGA-based Convolutional Neural Networks*, in *2017 27th International Conference on Field Programmable Logic and Applications（FPL）*,
pp.1-8, September, 2017.DOI.

[33] S.I. Venieris and C.-S.Bouganis, *fpgaConvNet:A Framework for Mapping Convolutional Neural Networks on FPGAs*, in *2016 IEEE 24th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp.40-47, Institute of Electrical and Electronics Engineers（IEEE）, May, 2016.DOI.

[34] Y.Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick等人，*Caffe:用于快速特征嵌入的卷积架构*，*arXiv:1408.5093* (2014)。

[35] Y.Guan, H. Liang, N. Xu, W. Wang, S. Shi, X. Chen et al., *Fp-dnn:一个将深度神经网络映射到具有rtl-hls混合模板的FPGA上的自动化框架*，*在2017年IEEE第25届现场可编程定制计算机（FCCM）年度国际研讨会上*，第152-159页，4月，2017。DOI.

[36] M.Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro等人, *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015.

[37] S.Hochreiter and J. Schmidhuber, *Long short-term memory*, *Neural Comput.***9** (Nove., 1997) 1735-1780.

[38] H.Sharma, J. Park, D. Mahajan, E. Amaro, J. K. Kim, C. Shao等, *From high-level deep neural models to fpgas*, in *Microarchitecture (MICRO), 2016 49th Annual IEEE/ACM International Symposium on*, pp.1-12, IEEE, 2016.

[39] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman and A. Schwartzman, *Jet-images âĂŤ deep learning edition*, *JHEP* **07** (2016) 069, [1511.05190]。

[40] D.Guest, J. Collado, P. Baldi, S. -C.Hsu, G. Urban and D. Whiteson, *Jet Flavor Classification in High-Energy Physics with Deep Neural Networks*, *Phys. Rev.* **D94** (2016) 112002, [1607.08633]。

[41] S.Macaluso and D. Shih, *Pulling Out All the Tops with Computer Vision and Deep Learning*, 1803.00107.

[42] K.Datta and A. J. Larkoski, *Novel Jet Observables from Machine Learning*, *JHEP* **03** (2018) 086, [1710.01305].

[43] A.Butter, G. Kasieczka, T. Plehn and M. Russell, *Deep-learned Top Tagging with a Lorentz Layer*, 1707.08966.

[44] G. Kasieczka, T. Plehn, M. Russell and T. Schell, *Deep-learning Top Taggers or The End of QCD?*, *JHEP* **05** (2017) 006, [1701.08784].

[45] P.T. Komiske, E. M. Metodiev and M. D. Schwartz, *Deep learning in color: towards automated quark/gluon jet discrimination*, *JHEP* **01** (2017) 110, [1612.01551].

[46] A.Schwartzman, M. Kagan, L. Mackey, B. Nachman and L. De Oliveira, *Image Processing, Computer Vision, and Deep Learning: New approaches to the analysis and physics interpretation of LHC events*, *J. Phys. Conf.Ser.* **762** (2016) 012035.

[47] J.M. Butterworth, A. R. Davison, M. Rubin and G. P. Salam, *Jet substructure as a new Higgs search channel at the LHC*, *Phys. Rev. Lett.***100** (2008) 242001, [0802.2470].

[48] L.阿斯奎斯等人，*大型强子对撞机的喷射物结构。实验回顾*, 1803.06991.

[49] J.Cogan, M. Kagan, E. Strauss and A. Schwarztman, *Jet-Images:Jet-Images: Computer Vision Inspired Techniques for Jet Tagging*, *JHEP* **02** (2015) 118, [1407.5675] 。

[50] J.Pearkes, W. Fedorko, A. Lister and C. Gay, *Jet Constituents for Deep Neural Network Based Top Quark Tagging*, 1704.02124.

[51] G. Louppe, K. Cho, C. Becot and K. Cranmer, *QCD-Aware Recursive Neural Networks for Jet Physics*, 1702.00748.

[52] CMS合作，A.M.Sirunyan等人，*搜索衰变到的低质量向量共振在$\sqrt{s}$=13TeV的质子-质子对撞中的夸克-反夸克对*，*JHEP* **01** （2018）097，[1710.00159]。

[53] CMS合作，A. M. Sirunyan等人，*Inclusive search for a highly boosted Higgs boson decaying to a bottom quark-antiquark pair*, *Phys. Rev. Lett.***120** (2018) 071802, [1709.05543].

[54] J.Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer et al., *The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations*, *JHEP* **07** (2014) 079, [1405.0301] 。

[55] R.D. Ball等人，《*LHC数据的粒子分布*》，*Nucl.***B867** (2013) 244-289, [1207.1303].

[56] T.Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten等, *An Introduction to PYTHIA 8.2*, *Comput.Phys. Commun.***191** (2015) 159-177, [1410.3012].

[57] P.Skands, S. Carrazza and J. Rojo, *Tuning PYTHIA 8.1: the Monash 2013 Tune*, *Eur.Phys. J.* **C74** (2014) 3024, [1404.5630].

[58] M.Cacciari, G. P. Salam and G. Soyez, *Fastjet用户手册*，*Eur.Phys. J.* **C72** (2012) 1896, [1111.6097].

[59] M.Cacciari and G. P. Salam, *Dispelling the n^3 myth for the k_t jet-finder*, *Phys. Lett.***B641** (2006) 57-61, [hep-ph/0512210].

[60] M.Cacciari, G. P. Salam and G. Soyez, *The Anti-k(t) jet clustering algorithm*, *JHEP* **04** (2008) 063, [0802.1189] 。

[61] D.Adams等人，《*走向对喷气结构中的相关性的理解*》，*Eur.Phys. J.* **C75** (2015) 409, [1504.00679].

[62] A.J. Larkoski, S. Marzani, G. Soyez and J. Thaler, *Soft Drop*, *JHEP* **05** (2014) 146, [1402.2657] 。

[63] A.J. Larkoski, G. P. Salam and J. Thaler, *Energy Correlation Functions for Jet Substructure*, *JHEP* **06** (2013) 108, [1305.0007].

[64] I.Moult, L. Necib and J. Thaler, *New Angles on Energy Correlation Functions*, *JHEP* **12** (2016) 153, [1609.07483] 。

[65] E.Coleman, M. Freytsis, A. Hinzmann, M. Narain, J. Thaler, N. Tran et al., *The importance of calorimetry for highly-boosted jet substructure*, *JINST* **13** (2018) T01003, [1709.08705] 。

[66] V.Nair and G. E. Hinton, *Rectified linear units improve restricted Boltzmann machines*, in *Proceedings of ICML*, vol. 27, pp.

[67] D.P. Kingma和J. Ba, *Adam: A method for stochastic optimization*, *CoRR* **abs/1412.6980** (2014) , [1412.6980] 。

[68] AWS，"Amazon EC2 P2 Instances." https://aws.amazon.com/ec2/instance-types/p2/, 2018。

[69] S.Han, H. Mao and W. J. Dally, *Deep compression:用剪枝、训练有素的量化和胡夫曼编码压缩深度神经网络*, *CoRR* **abs/1510.00149** (2015) , [1510.00149] 。

[70] Y.Cheng, D. Wang, P. Zhou and T. Zhang, *A survey of model compression and acceleration for deep neural networks*, *CoRR* **abs/1710.09282** (2017) , [1710.09282] 。

[71] Y.LeCun, J. S. Denker and S. A. Solla, *Optimal brain damage*, in *Advances in Neural Information Processing Systems 2* (D. S. Touretzky, ed.), pp.Morgan-Kaufmann, 1990.

[72] C.Louizos, M. Welling and D. P. Kingma, *Learning Sparse Neural Networks through L_0 Regularization*, *ArXiv e-prints* (Dec., 2017) , [1712.01312] 。

[73] R.Rigamonti, A. Sironi, V. Lepetit and P. Fua, *Learning separable filters*, in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp.2754-2761, June, 2013.DOI.

[74] E.L. Denton, W. Zaremba, J. Bruna, Y. LeCun and R. Fergus, *Exploiting linear structure within convolutional networks for efficient evaluation*, in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence and K. Q. Weinberger, eds., pp. 1269-1277.Curran Associates, Inc., 2014.

[75] M.Jaderberg, A. Vedaldi and A. Zisserman, *Speeding up convolutional neural networks with low rank expansions*, *CoRR* **abs/1405.3866** (2014) , [1405.3866] 。

[76] M.Denil, B. Shakibi, L. Dinh, M. A. Ranzato and N. de Freitas, *Predicting parameters in deep learning*, in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z.Ghahramani and K. Q. Weinberger, eds.), pp.2148-2156.Curran Associates, Inc., 2013.

[77] T.N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy and B. Ramabhadran, *Low-rank matrix factorization for deep neural network training with high-dimensional output targets*, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.6655-6659, May, 2013。DOI.

[78] T.S. Cohen and M. Welling, *Group Equivariant Convolutional Networks*, *ArXiv e-prints* (Feb., 2016) , [1602.07576] 。

[79] C.Buciluaˇ, R. Caruana and A. Niculescu-Mizil, *Model compression*, in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, (New York, NY, USA), pp.535-541, ACM, 2006.DOI.

[80] S.Han, J. Pool, J. Tran and W. J. Dally, *Learning both weights and connections for efficient neural networks*, *CoRR* **abs/1506.02626** (2015) , [1506.02626] 。

[81] A.Y. Ng, *特征选择，l1与l2正则化，以及旋转不变性*，*在第二十一届国际机器学习会议论文集*, ICML '04，（纽约，美国），第78-，ACM，2004。DOI.

[82] Y.Gong, L. Liu, M. Yang and L. D. Bourdev, *Compressing deep convolutional networks using vector quantization*, *CoRR* **abs/1412.6115** (2014) , [1412.6115] 。

[83] J.Wu, C. Leng, Y. Wang, Q. Hu and J. Cheng, *Quantized convolutional neural networks for mobile devices*, in *2016 IEEE Conference on Computer Vision and Pattern Recognition （CVPR）*, pp.4820-4828, June, 2016.DOI.

[84] V.Vanhoucke, A. Senior and M. Z. Mao, *Improving the speed of neural networks on cpus*, in *Deep Learning and Unsupervised Feature Learning Workshop, NIPS 2011*, 2011.

[85] S.Gupta, A. Agrawal, K. Gopalakrishnan and P. Narayanan, *Deep learning with limited numerical precision*, *CoRR* **abs/1502.02551** (2015) , [1502.02551] 。

[86] M.Courbariaux, Y. Bengio and J.-P.David, *Binaryconnect:在传播过程中用二进制权重训练深度神经网络*, 《*神经信息处理系统进展*》*28期*（C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama and R. Garnett, eds.）, 第3123-3131页。Curran Associates, Inc., 2015.

[87] I.Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv and Y. Bengio, *Binarized neural networks*, in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon and R. Garnett, eds.) , pp.4107-4115.Curran Associates, Inc., 2016.

[88] M.Rastegari, V. Ordonez, J. Redmon and A. Farhadi, *Xnor-net:使用二元卷积神经网络的Imagenet分类*, *CoRR* **abs/1603.05279** (2016) , [1603.05279] 。

[89] P.Merolla, R. Appuswamy, J. V. Arthur, S. K. Esser and D. S. Modha, *Deep neural networks are robust to weight binarization and other non-linear distortions*, *CoRR* **abs/1606.01981** (2016) , [1606.01981] 。

[90] Xilinx, *7系列DSP48E1切片用户指南*。

[91] C.Zhu, S. Han, H. Mao and W. J. Dally, *Trained ternary quantization*, *CoRR* **abs/1612.01064** (2016) , [1612.01064] 。

[92] S.Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz等, *EIE: Efficient Inference Engine on Compressed Deep Neural Network*, *ArXiv e-prints* (Feb., 2016) , [1602.01528] 。

[93] M.Paganini, L. de Oliveira and B. Nachman, *Accelerating Science with Generative Adversarial Networks:An Application to 3D Particle Showers in Multilayer Calorimeters*, Phys. *Rev. Lett.***120** (2018) 042003, [1705.02355].

[94] D.伯格，"微软公布了用于实时AI的脑波项目。" https://www.microsoft.com/en-us/research/blog/microsoft-unveils-project-brainwave/，2017年。

[95] A.M. Caulfield, E. S. Chung, A. Putnam, H. Angepat, D. Firestone, J. Fowers等人， *可配置的云*, 在*Micro卷37。Issue: 3*, pp. 52-61, IEEE, 2017.

[96] S.Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz等, *EIE: efficient inference engine on compressed deep neural network*, *CoRR* **abs/1602.01528** (2016) , [1602.01528] 。