# 一种基于 FPGA 的图神经网络加速器解决方案

# （白皮书编号：WP024）

得益于大数据的兴起和计算能力的快速提升，机器学习技术近年来经历了革命性的发展。诸如图像分类、语音识别和自然语言处理等机器学习任务，都是对具有一定大小、维度和有序排列的欧几里得数据进行处理。然而，在许多现实场景中，数据是由复杂的非欧几里得数据（例如图形）表示的。这些图形不仅包含数据，还包含数据之间的依赖关系，例如社交网络、蛋白质分子结构、电子商务平台中的客户数据等。数据复杂性的提升给传统的机器学习算法设计及其实现技术带来了严峻的挑战。在这种情况下，许多全新的基于图形的机器学习算法或图神经网络（GNN）不断在学术界和工业界涌现。

GNN 对计算能力和存储有非常高的要求，而且其算法的软件实现效率非常低。因此，业界对 GNN 的硬件加速有着非常迫切的需求。尽管传统的卷积神经网络（CNN）硬件加速有很多种解决方案，但 GNN 的硬件加速还没有得到充分的讨论和研究。在撰写本白皮书时，谷歌（Google）和百度（Baidu）都无法搜索到关于 GNN 硬件加速的中文研究资料。本白皮书的写作动机是将国外最新的 GNN 算法、对加速技术的研究以及对基于现场可编程逻辑门阵列（FPGA）的 GNN 加速技术的探讨相结合，并以概述的形式呈现给读者。

## 对图神经网络（GNN）的介绍

在宏观层面上，GNN 的架构与传统 CNN 有很多相似之处，诸如卷积层、池化、激活函数、机器学习处理器（MLP）、全连接层（FC layer）等模块，这些都可以应用到 GNN。下图展示了一个相对简单的 GNN 架构。

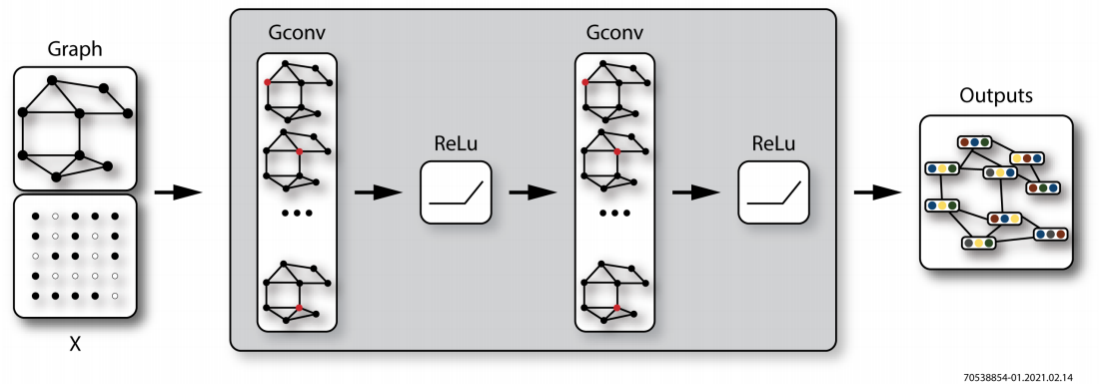


**图 1：*典型的 GNN 架构（来源：https://arxiv.org/abs/1901.00596）***

但是，GNN 中的图形数据卷积计算与传统 CNN 中的二维卷积计算不同。以下图为例，红色目标节点的卷积计算过程如下所示：

1、图卷积 - 使用近邻函数对周围节点的特征进行采样，并计算平均值。相邻节点的数量是不确定且无序的（非欧几里得数据）

2、二维卷积——使用卷积核对周围节点的特征进行采样，并计算加权平均值。相邻节点的数量是确定且有序的（欧几里得数据）

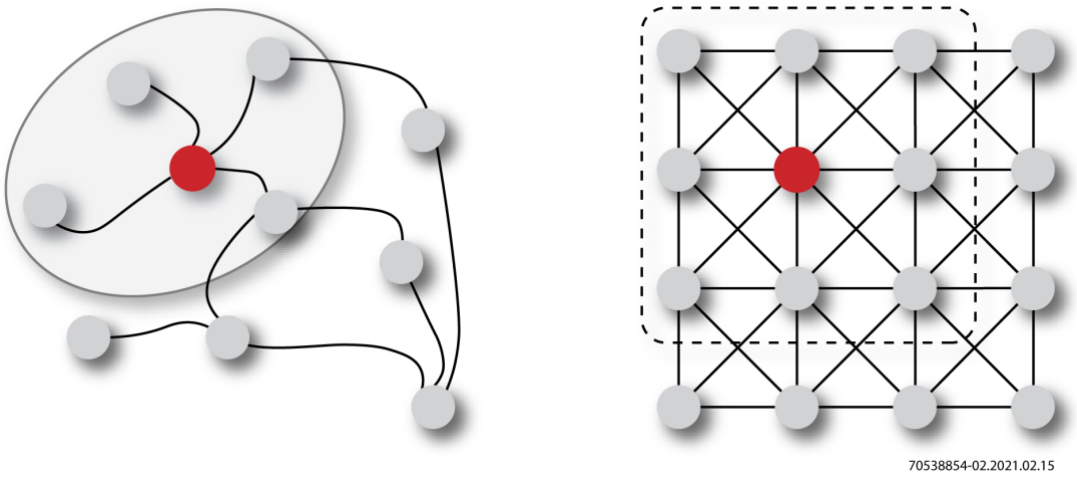


**图 2：*图卷积和二维卷积（来源：https://arxiv.org/abs/1901.00596）***

## 对 GraphSAGE 算法的介绍

学术界对 GNN 算法进行了大量的研究和探讨，提出了相当多的创新实现方法。其中，由斯坦福大学（Stanford University）于 2017 年提出的 GraphSAGE 是一种归纳表示学习算法，用于预测大规模图中动态的、全新的、未知的节点类型，还专门针对节点数量庞大、节点特征丰富的图进行了优化。如下图所示，GraphSAGE 算法的计算过程可以分为三个主要步骤：

1、相邻节点采样——用于降低复杂性，一般采样两层，每层采样几个节点。

2、聚合——用于嵌入目标节点，即图的低维向量表示。

3、预测——使用嵌入作为全连接层的输入，以预测目标节点 d 的标签。

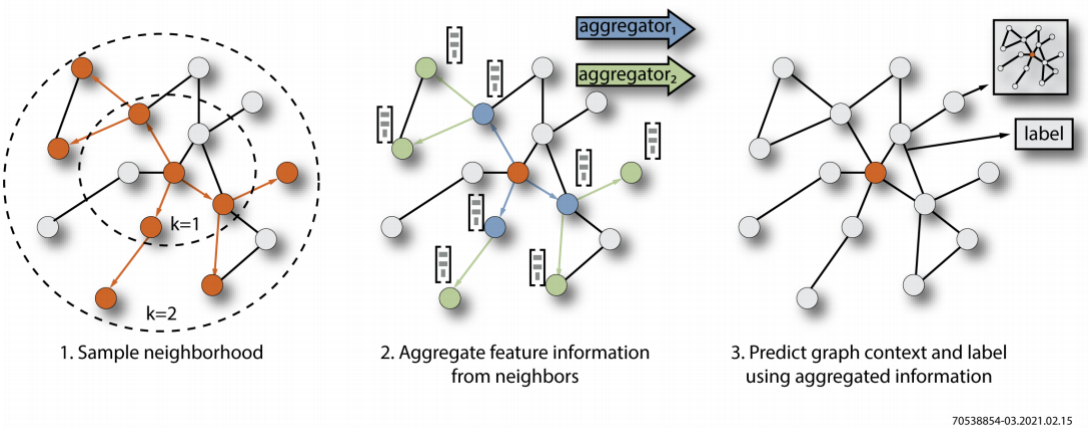


**图 3：*GraphSAGE 算法的可视化表示（来源：***

***http://snap.stanford.edu/graphsage）***

1.Sample neighborhood

1、样本邻域

2.Aggregate feature information from neighbors

2、聚合来自邻域的特征信息

3.Predict graph context and label using aggregated information

3、利用聚合信息预测图形情况和标签

为了在 FPGA 中实现 GraphSAGE 算法加速，必须了解其数学模型，以便将算法映射到不同的逻辑模块。下图所示的代码说明了该算法的数学过程。

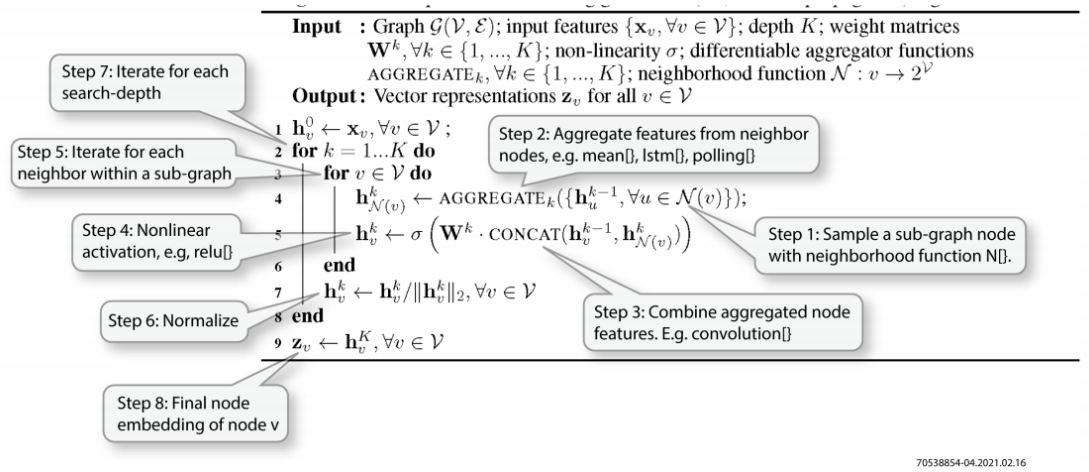


**图 4：*GraphSAGE 算法的数学模型（来源：***

***http://snap.stanford.edu/graphsage*）**

Step 1: Sample a sub-graph node with neighborhood function N[].

步骤 1：使用近邻函数 N[]对子图节点进行采样。

Step 2: Aggregate features from neighbor nodes, e.g. mean[], lstm[], polling[]

步骤 2：聚合相邻节点的特征，例如 mean[]、lstm[]、polling[]

Step3: Combine aggregated node features. E.g. convolution[]

步骤 3：合并聚合的节点特征。例如卷积[]

Step 4: Nonlinear activation, e.g, relu[]

步骤 4：非线性激活，例如 relu[]

Step 5: Iterate for each neighbor with a sub-graph

步骤 5：使用子图迭代每个邻域

Step 6: Normalize

步骤 6：标准化

Step 7: Iterate for each search-depth

步骤 7：对每个深度搜索进行迭代

Step 8: Final node embedding of node v

步骤 8：节点 v 的最终节点嵌入

对于每个要处理的目标节点 xv，GraphSAGE 算法都会执行以下操作：

1、通过近邻采样函数 N(v)对子图中的节点进行采样。

2、聚合要采样的相邻节点的特征。聚合函数可以是 mean()、lstm()或 polling()等。

3、将聚合结果与上一次迭代的输出表示合并起来，并使用 Wk 进行卷积。

4、对卷积结果进行非线性处理。

5、多次迭代以结束当前第 k 层的所有相邻节点的处理。

6、对第 k 层迭代的结果进行标准化处理。

7、多次迭代以结束对所有 K 层采样深度的处理。

8、将最终的迭代结果 zv 嵌入到输入节点 xv。

## GNN 加速器设计所面临的挑战

GNN 算法涉及大量的矩阵计算和存储访问操作。在传统的 x86 架构服务器上运行这种算法的效率是非常低的，表现为速度慢、能耗高等。

新型图形处理器（GPU）的应用可以显著提高 GNN 的计算速度与能效比。但是，GPU 在存储可扩展性方面存在短板，使其无法处理图形中的海量节点。GPU 的指令执行方式也会导致计算延迟过大和不确定性；因此，它不适用于需要实时计算图形的场景。

上面提到的各种设计挑战，使得业界迫切需要一种能够支持高并发、实时计算，拥有巨大存储容量和带宽，并可扩展到数据中心的 GNN 加速解决方案。

## 基于 FPGA 设计方案的 GNN 加速器

Achronix 的 Speedster®7t 系列 FPGA 产品（以及该系列的第一款器件

AC7t1500）是针对数据中心和机器学习工作负载进行了优化的高性能 FPGA 器件，消除了基于中央处理器（CPU）、GPU 和传统 FPGA 的解决方案中存在的若干性能瓶颈。Speedster7t 系列 FPGA 产品采用了台积电（TSMC）的 7nm FinFET 工艺，其架构采用了一种革命性的全新二维片上网络（NoC）、独创的机器学习处理器矩阵（MLP），并采用高带宽 GDDR6 控制器、400G 以太网和 PCI Express Gen5 接口，在确保 ASIC 级性能的同时，它为用户提供了灵活的硬件可编程性。下图展示了高性能 FPGA 器件 Speedster7t1500 的架构。


**图 5：*Achronix 高性能 FPGA 器件 Speedster AC7t1500 的架构***

上述特点使 Achronix Speedster7t1500 器件成为应对在 GNN 加速器设计中面临的各种挑战的完美解决方案。

***表 1：GNN 设计面临的挑战和 Achronix Speedster7t1500 FPGA 器件提供的解决方案***

| **GNN 设计所面临的挑战** | **Speedster AC7t1500 器件提供的解决方案** |
|---|---|
| 高速矩阵运算 | 机器学习处理器（MLP） |
| 高带宽和低延迟存储 | LRAM+BRAM+GDDR6+DDR4。 |

| | |
|---|---|
| 高并发和低延迟计算 | FPGA 使用可编程逻辑电路，以确保在硬件层面进行低并发和高并发延迟计算。 |
| 存储扩展 | 基于 4×400 Gbps 的 RDMA 确保在数据中心以极低的延迟扩展存储访问。 |
| 算法不断演进 | FPGA 中的可编程逻辑确保算法可以在硬件层面进行升级和重新配置。 |
| 复杂的设计 | 丰富的硬 IP 减少开发时间、降低复杂性，NoC 简化模块之间的互连并改善时序 |

## GNN 加速器顶层架构

此 GNN 加速器是为 GraphSAGE 算法设计的，但是它的设计也可以应用于其他类似的 GNN 算法加速。其顶层架构如下图所示。

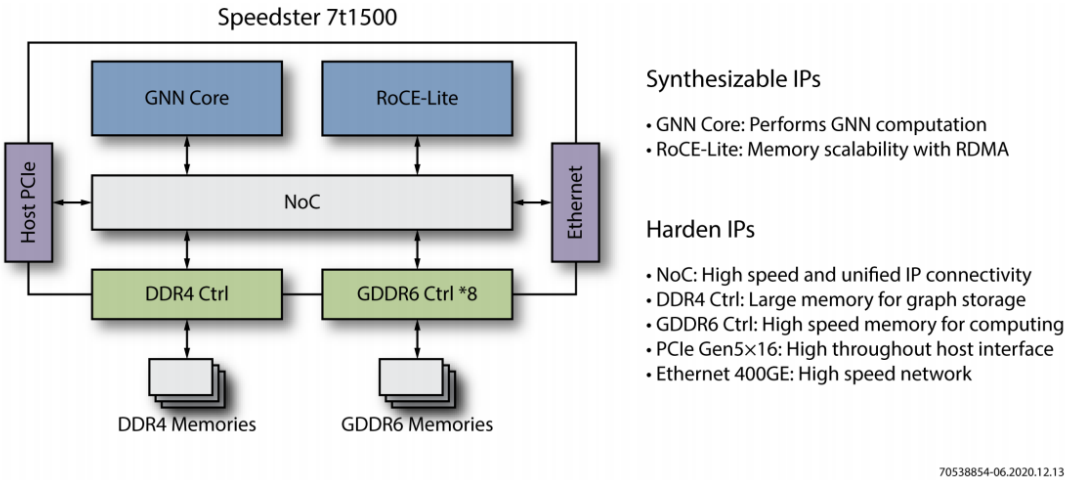


**图 6：*GNN 加速器顶层架构***

Synthesizable IPs

可综合的 IP

GNN Core: Preforms GNN computation

GNN 内核：执行 GNN 计算

RoCE-Lite: Memory scalability with RDMA

RoCE-Lite：采用 RDMA 的存储可扩展性

Harden IPs

硬化 IP

NoC: High speed and unified IP connectivity

NoC：高速、统一的 IP 连接

DDR4 Ctrl: Large memory for graph storage

DDR4 Ctrl：用于图形存储的大存储容量

GDDR6 Ctrl: High speed memory for computing

GDDR6 Ctrl：用于计算的高速存储

PCIe Gen5×16: High throughout host interface

PCIe Gen5×16：高吞吐量的主机接口

Ethernet 400GE: High speed network

以太网 400GE：高速网络

该架构由以下模块组成：

- 图中的 GNN 内核是算法实现的核心部分（详情如下）。
- RoCE-Lite 是 RDMA 协议的轻量级版本，用于通过高速以太网进行远程存储访问，以支持海量节点的图计算。
- 400GE 以太网控制器用于承载 RoCE-Lite 协议。
- GDDR6 存储器用于存储 GNN 处理过程中所需的高速访问数据（DDR4 作为备用大容量存储器）。该存储器用于存储访问频率相对较低的数据，例如待预处理的图形数据。
- PCIe Gen5 ×16 接口提供高速主机接口，用于与服务器软件进行数据交互。

上述所有模块均通过具有高带宽的 NoC 实现互连。

## GNN 内核微架构

在开始讨论 GNN 内核的微架构之前，有必要先回顾一下 GraphSAGE 算法。

其内层循环的聚合和合并（包括卷积）占据了该算法的大部分计算和存储访问。通过研究，我们得出这两个步骤的特点，具体如下。

***表 2：GNN 算法中聚合和合并操作的对比（来源：https://arxiv.org/abs/1908.10834）***

| 步骤 | 聚合操作 | 合并操作 |
|---|---|---|
| 存储访问方式 | 间接访问，不规则 | 直接访问，规则 |
| 数据重用 | 低 | 高 |
| 计算模式 | 动态，不规则 | 静态，规则 |
| 计算量 | 低 | 高 |
| 性能瓶颈 | 存储 | 计算 |

可以看出，聚合操作和合并操作在计算和存储访问模式上有着完全不同的需求。聚合操作涉及相邻节点的采样。然而，图形是一种非欧几里得数据类型——它的大小和维度是不确定且无序，矩阵稀疏，节点位置随机。因此，存储访问是不规则的，并且难以重复利用数据。

在合并操作中，输入数据是聚合结果（节点的低维表示）和权重矩阵。它的大小和维度是固定的，具有线性存储位置。因此对存储访问没有挑战，但是矩阵的计算量非常大。

基于上述分析，我们决定在 GNN 内核加速器设计中选择使用两种不同的硬件结构来分别处理聚合和合并操作（如下图示）：

- 聚合器——通过单指令多数据（SIMD）处理器阵列，对图形相邻节点进行采样和聚合。单指令可以预定义为 mean()平均值计算，或其他适用的聚合函数；多数据是指单次 mean()均值计算中需要多个相邻节点的特征数据作为输入，这些数据来自子图采样器。SIMD 处理器阵列通过调度器 Agg Scheduler 进行负载平衡。子图采样器通过 NoC 从 GDDR6 或 DDR4 读回的邻接矩阵和节点特征数据 h0v 分别缓存在邻接列表缓冲区（Adjacent List Buffer）和节点特征缓冲区（Node Feature Buffer）。聚合的结果 hkN(v)存

储在聚合缓冲区（Aggregation Buffer）中。

- 合并器——通过脉动矩阵 PE 对聚合结果进行卷积运算。卷积核是 Wk 权重矩阵。卷积结果由 ReLU 激活函数进行非线性处理，同时也存储在 Partial Sum Buffer 中，以用于下一轮迭代。

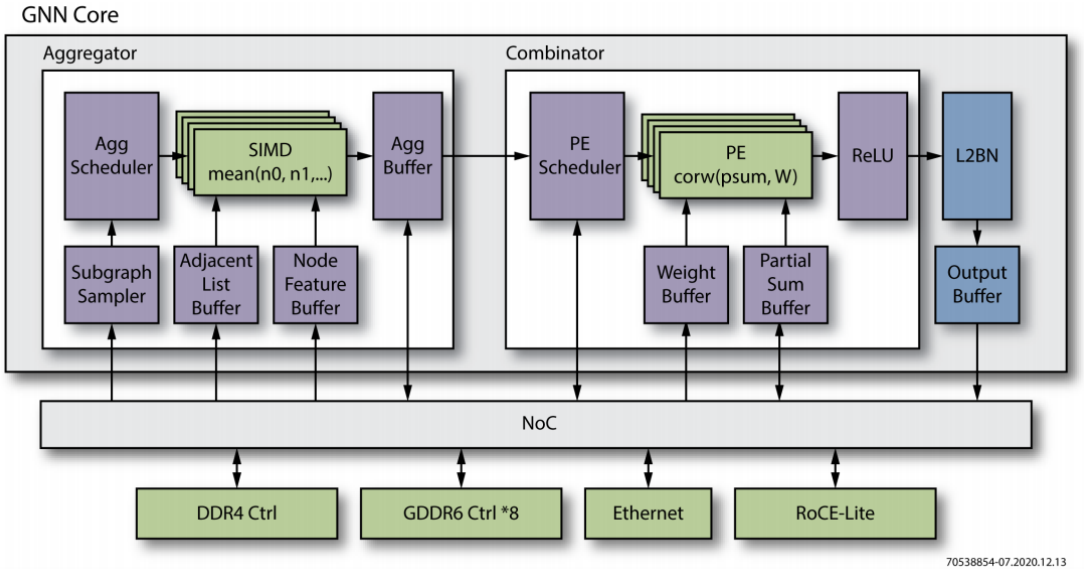


**图 7：*GNN 内核功能框图***

合并结果经过 L2BN 标准化处理后，即为最终的节点表示 hkv。在一个典型的节点分类预测应用中，节点表示 hkv 可以通过一个全连接层（FC）来获取节点的分类标签。这个过程是传统的机器学习处理方法之一，在 GraphSAGE 文献资料中没有体现，这个功能也没有包含在这个架构中。

## 结论

本白皮书探讨了 GraphSAGE GNN 算法的数学原理，并从多个角度分析了 GNN 加速器设计中的技术挑战。通过分析问题并在架构层面逐一解决，提出了一种架构，利用 Achronix Speedster7t AC7t1500 FPGA 器件提供的具有竞争性的优势，创建了一种高度可扩展的、能够提供卓越性能的 GNN 加速解决方案。

有关 Speedster7t 系列 FPGA 器件的更多信息，请访问 www.achronix.com。



## 免责声明