

一種基於 FPGA 的圖神經網路加速器解決方案

■文：Achronix

得益於大資料的興起和計算能力的快速提升，機器學習技術近年來經歷了革命性的發展。諸如圖像分類、語音辨識和自然語言處理等機器學習任務，都是對具有一定大小、維度和有序排列的歐幾裡得資料進行處理。然而，在許多現實場景中，資料是由複雜的非歐幾裡得資料（例如圖形）表示的。這些圖形不僅包含資料，還包含資料之間的依賴關係，例如社交網路、蛋白質分子結構、電子商務平臺中的客戶資料等。資料複雜性的提升給傳統的機器學習演算法設計及其實現技術帶來了嚴峻的挑戰。在這種情況下，許多全新的基於圖形的機器學習演算法或圖神經網路 (GNN) 不斷在學術界和工業界湧現。

GNN 對計算能力和存儲有非常高的要求，而且其演算法的軟體實現效率非常低。因此，業界對 GNN 的硬體加速有著非常迫切的需求。儘管傳統的卷積神經網路 (CNN) 硬體加速有很多種解決方案，但 GNN 的硬體加速還沒有得到充分的討論和研究。

在撰寫本白皮書時，谷歌 (Google) 和百度 (Baidu) 都無法搜索到關於 GNN 硬體加速的中文研究資料。本白皮書的寫作動機是將國外最新的 GNN 演算法、對加速技術的研究以及對基於現場可程式設計邏輯閘陣列 (FPGA) 的 GNN 加速技術的探討相結合，並以概述的形式呈現給讀者。

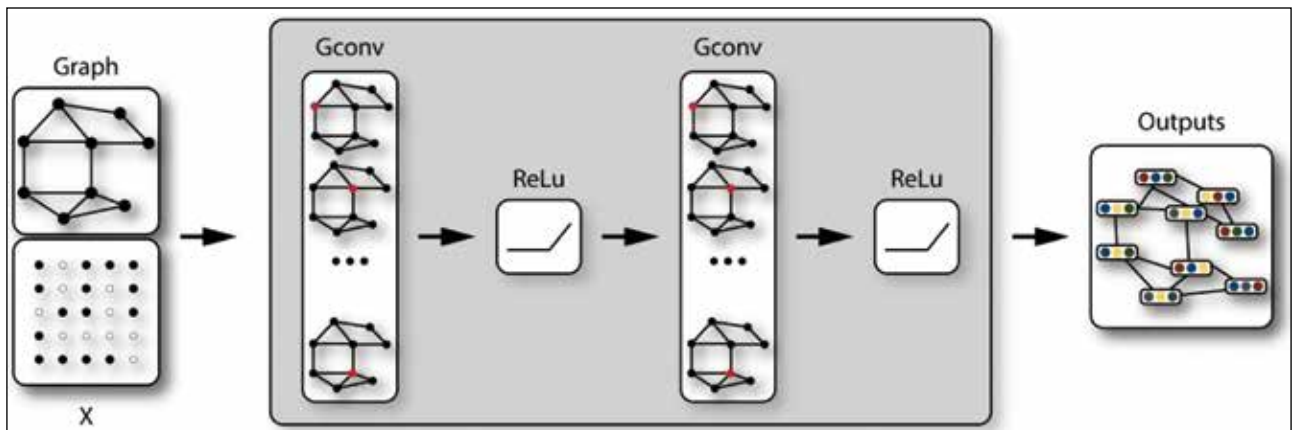
對圖神經網路 (GNN) 的介紹

在宏觀層面上，GNN 的架構與傳統 CNN 有很多相似之處，諸如卷積層、池化、啟動函數、機器學習處理器 (MLP)、全連接層 (FC layer) 等模組，這些都可以應用到 GNN。圖 1 展示了一個相對簡單的 GNN 架構。

但是，GNN 中的圖形資料卷積計算與傳統 CNN 中的二維卷積計算不同。以圖 2 為例，紅色目標節點的卷積計算過程如下所示：

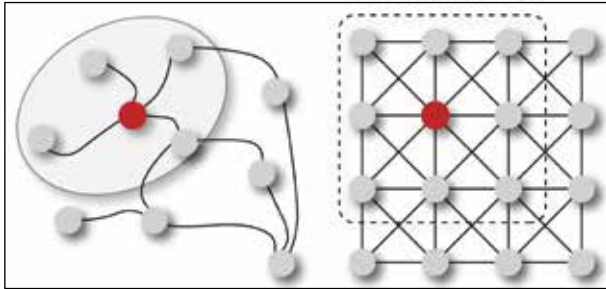
1、圖卷積 - 使用近鄰函數對周圍節點的特徵進行

圖 1：典型的 GNN 架構



資料來源：<https://arxiv.org/abs/1901.00596>

圖 2：圖卷積和二維卷積



資料來源：<https://arxiv.org/abs/1901.00596>

採樣，並計算平均值。相鄰節點的數量是不確定且無序的 (非歐幾裡得數據)

2、二維卷積——使用卷積核對周圍節點的特徵進行採樣，並計算加權平均值。相鄰節點的數量是確

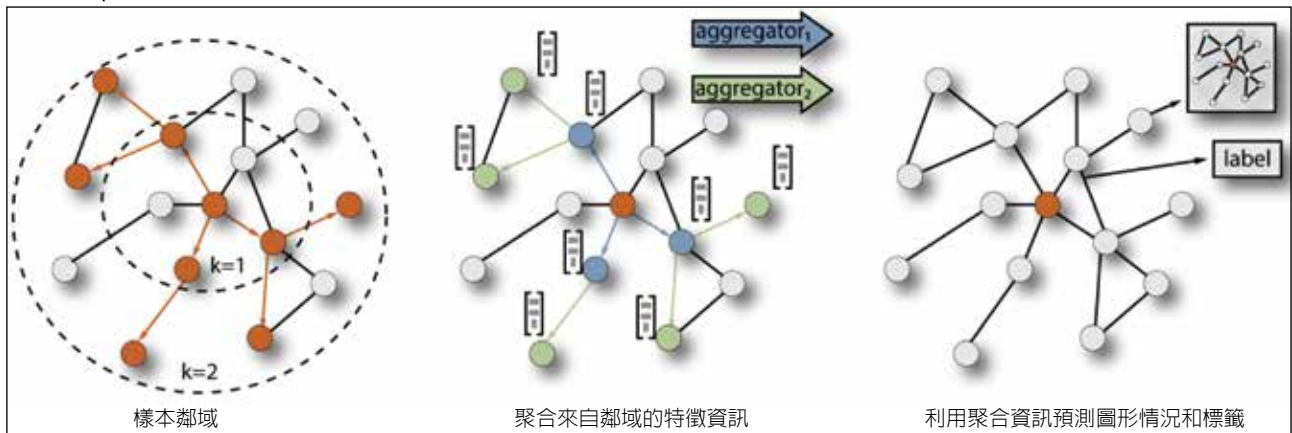
定且有序的 (歐幾裡得資料)

對 GraphSAGE 演算法的介紹

學術界對 GNN 演算法進行了大量的研究和探討，提出了相當多的創新實現方法。其中，由史丹佛大學 (Stanford University) 於 2017 年提出的 GraphSAGE 是一種歸納表示學習演算法，用於預測大規模圖中動態的、全新的、未知的節點類型，還專門針對節點數量龐大、節點特徵豐富的圖進行了優化。如圖 3 所示，GraphSAGE 演算法的計算過程可以分為三個主要步驟：

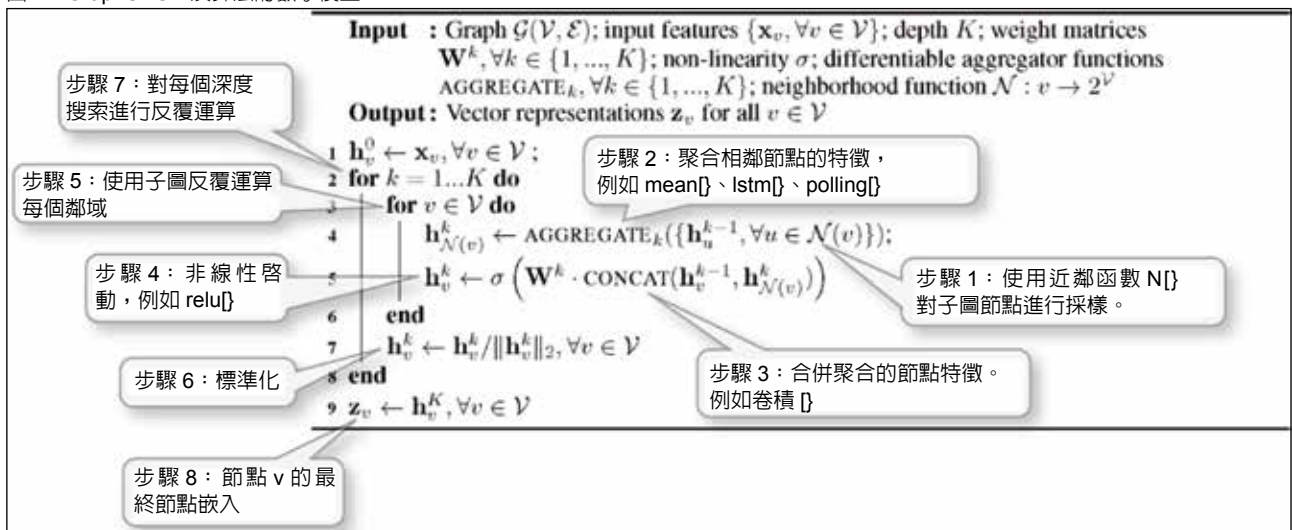
1、相鄰節點採樣——用於降低複雜性，一般採樣兩

圖 3：GraphSAGE 演算法的視覺化表示



資料來源：<http://snap.stanford.edu/graphsage>

圖 4：GraphSAGE 演算法的數學模型



資料來源：<http://snap.stanford.edu/graphsage>

層，每層採樣幾個節點。

- 2、聚合——用於嵌入目標節點，即圖的低維向量表示。
- 3、預測——使用嵌入作為全連接層的輸入，以預測目標節點 d 的標籤。

為了在 FPGA 中實現 GraphSAGE 演算法加速，必須瞭解其數學模型，以便將演算法映射到不同的邏輯模組。圖 4 所示的代碼說明了該演算法的數學過程。

對於每個要處理的目標節點 xv ，GraphSAGE 演算法都會執行以下操作：

- 1、通過近鄰採樣函數 $N(v)$ 對子圖中的節點進行採樣。
- 2、聚合要採樣的相鄰節點的特徵。彙總函式可以是 $mean()$ 、 $lstm()$ 或 $polling()$ 等。
- 3、將聚合結果與上一次反覆運算的輸出表示合併起來，並使用 W_k 進行卷積。
- 4、對卷積結果進行非線性處理。
- 5、多次反覆運算以結束當前第 k 層的所有相鄰節點的處理。
- 6、對第 k 層反覆運算的結果進行標準化處理。
- 7、多次反覆運算以結束對所有 K 層採樣深度的處理。
- 8、將最終的反覆運算結果 zv 嵌入到輸入節點 xv 。

GNN 加速器設計所面臨的挑戰

GNN 演算法涉及大量的矩陣計算和存儲訪問操作。在傳統的 x86 架構伺服器上運行這種演算法的效率是非常低的，表現為速度慢、能耗高等。

新型圖形處理器 (GPU) 的應用可以顯著提高

GNN 的計算速度與能效比。但是，GPU 在存儲可擴展性方面存在短板，使其無法處理圖形中的海量節點。GPU 的指令執行方式也會導致計算延遲過大和不確定性；因此，它不適用於需要即時計算圖形的場景。

上面提到的各種設計挑戰，使得業界迫切需要一種能夠支援高併發、即時計算，擁有巨大存儲容量和頻寬，並可擴展到資料中心的 GNN 加速解決方案。

基於 FPGA 設計方案的 GNN 加速器

Achronix 的 Speedster 7t 系列 FPGA 產品 (以及該系列的第一款器件 AC7t1500) 是針對資料中心和機器學習工作負載進行了優化的高性能 FPGA 器件，消除了基於中央處理器 (CPU)、GPU 和傳統 FPGA 的解決方案中存在的若干性能瓶頸。Speedster7t 系列 FPGA 產品採用了台積電 (TSMC) 的 7nm FinFET 工藝，其架構採用了一種革命性的全新二維片上網路 (NoC)、獨創的機器學習處理器矩陣 (MLP)，並採用高頻寬 GDDR6 控制器、400G 乙太網和 PCI Express Gen5 介面，在確保 ASIC 級性能的同時，它為使用者提供了靈活的

圖 5：Achronix 高性能 FPGA 器件 Speedster AC7t1500 的架構

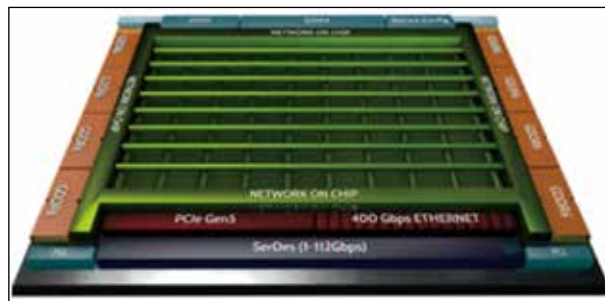


表 1：GNN 設計面臨的挑戰和 Achronix Speedster7t1500 FPGA 器件提供的解決方案

GNN 設計所面臨的挑戰	Speedster AC7t1500 器件提供的解決方案
高速矩陣運算	機器學習處理器 (MLP)
高頻寬和低延遲存儲	LRAM+BRAM+GDDR6+DDR4。
高併發和低延遲計算	FPGA 使用可程式設計邏輯電路，以確保在硬體層面進行低併發和高併發延遲計算。
存儲擴展	基於 4×400 Gbps 的 RDMA 確保在資料中心以極低的延遲擴展存儲訪問。
演算法不斷演進	FPGA 中的可程式設計邏輯確保演算法可以在硬體層面進行升級和重新配置。
複雜的設計	豐富的硬 IP 減少開發時間、降低複雜性，NoC 簡化模組之間的互連並改善時序

硬體可程式設計性。圖 5 展示了高性能 FPGA 器件 Speedster7t1500 的架構。

上述特點使 Achronix Speedster7t1500 器件成為應對在 GNN 加速器設計中面臨的各種挑戰的完美解決方案。

GNN 加速器頂層架構

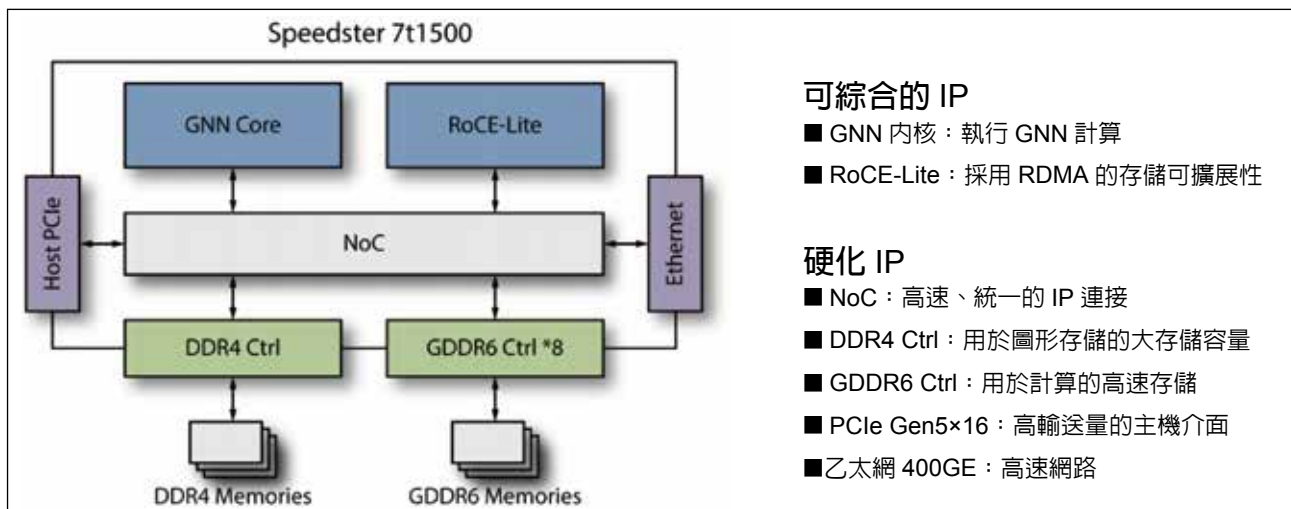
此 GNN 加速器是為 GraphSAGE 演算法設計的，但是它的設計也可以應用於其他類似的 GNN 演算法加速。其頂層架構如圖 6 所示。

該架構由以下模組組成：

- 圖中的 GNN 內核是演算法實現的核心部分 (詳情如下)。
- RoCE-Lite 是 RDMA 協議的羽量級版本，用於通過高速乙太網進行遠端存放訪問，以支持海量節點的圖計算。
- 400GE 乙太網控制器用於承載 RoCE-Lite 協定。
- GDDR6 記憶體用於存儲 GNN 處理過程中所需的高速訪問資料 (DDR4 作為備用大量存放區)。該記憶體用於存儲訪問頻率相對較低的資料，例如待預處理的圖形資料。
- PCIe Gen5×16 介面提供高速主機介面，用於與伺服器軟體進行資料交互。

上述所有模組均通過具有高頻寬的 NoC 實現互連。

圖 6：GNN 加速器頂層架構



GNN 內核微架構

在開始討論 GNN 內核的微架構之前，有必要先回顧一下 GraphSAGE 演算法。其內層迴圈的聚合和合併 (包括卷積) 佔據了該演算法的大部分計算和存儲訪問。通過研究，我們得出這兩個步驟的特點，具體如下。

表 2：GNN 演算法中聚合和合併操作的對比

步驟	聚合操作	合併操作
存儲訪問方式	間接訪問，不規則	直接訪問，規則
資料重用	低	高
計算模式	動態，不規則	靜態，規則
計算量	低	高
性能瓶頸	存儲	計算

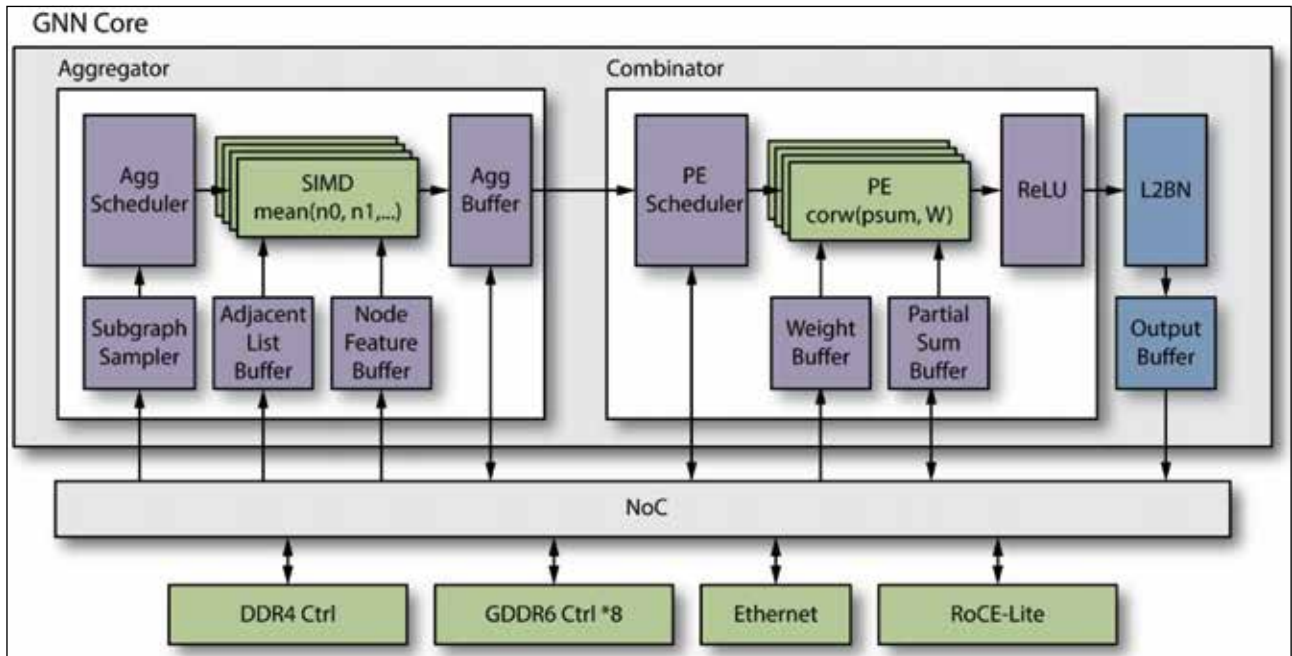
資料來源：<https://arxiv.org/abs/1908.10834>

可以看出，聚合操作和合併操作在計算和存儲訪問模式上有著完全不同的需求。聚合操作涉及相鄰節點的採樣。然而，圖形是一種非歐幾裡得資料類型——它的大小和維度是不確定且無序，矩陣稀疏，節點位置隨機。因此，存儲訪問是不規則的，並且難以重複利用資料。

在合併操作中，輸入資料是聚合結果 (節點的低維表示) 和權重矩陣。它的大小和維度是固定的，具有線性存儲位置。因此對存儲訪問沒有挑戰，但是矩陣的計算量非常大。

基於上述分析，我們決定在 GNN 內核加速器設計中選擇使用兩種不同的硬體結構來分別處理聚

圖 7：GNN 內核功能框圖



合和合併操作 (如圖 7 示)：

- 聚合器——通過單指令多資料 (SIMD) 處理器陣列，對圖形相鄰節點進行採樣和聚合。單指令可以預定義為 $\text{mean}()$ 平均值計算，或其他適用的彙總函式；多資料是指單次 $\text{mean}()$ 均值計算中需要多個相鄰節點的特徵資料作為輸入，這些資料來自子圖採樣器。SIMD 處理器陣列通過調度器 Agg Scheduler 進行負載平衡。子圖採樣器通過 NoC 從 GDDR6 或 DDR4 讀回的鄰接矩陣和節點特徵資料 h_0v 分別緩存在鄰接列表緩衝區 (Adjacent List Buffer) 和節點特徵緩衝區 (Node Feature Buffer)。聚合的結果 $h_kN(v)$ 存儲在聚合緩衝區 (Aggregation Buffer) 中。
- 合併器——通過脈動矩陣 PE 對聚合結果進行卷積運算。卷積核是 W_k 權重矩陣。卷積結果由 ReLU 啓動函數進行非線性處理，同時也存儲在 Partial Sum Buffer 中，以用於下一輪反覆運算。

合併結果經過 L2BN 標準化處理後，即為最終的節點表示 hkv 。在一個典型的節點分類預測應用中，節點表示 hkv 可以通過一個全連接層 (FC) 來獲取節點的分類標籤。這個過程是傳統的機器學習處

理方法之一，在 GraphSAGE 文獻資料中沒有體現，這個功能也沒有包含在這個架構中。

結論

本白皮書探討了 GraphSAGE GNN 演算法的數學原理，並從多個角度分析了 GNN 加速器設計中的技術挑戰。通過分析問題並在架構層面逐一解決，提出了一種架構，利用 Achronix Speedster7t AC7t1500 FPGA 器件提供的具有競爭性的優勢，創建了一種高度可擴展的、能夠提供卓越性能的 GNN 加速解決方案。

有關 Speedster7t 系列 FPGA 器件的更多資訊，請訪問 www.achronix.com。 