计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2021.20210166
58(6):1204-1229,2021

图神经网络加速结构综述

李 涵[1,2] 严明玉[1,2] 吕征阳[1,2] 李文明[1] 叶笑春[1] 范东睿[1,2] 唐志敏[1,2]
[1](计算机体系结构国家重点实验室(中国科学院计算技术研究所) 北京 100190)
[2](中国科学院大学 北京 100049)
(lihan-ams@ict.ac.cn)

Survey on Graph Neural Network Acceleration Architectures

Li Han[1,2], Yan Mingyu[1,2], Lü Zhengyang[1,2], Li Wenming[1], Ye Xiaochun[1], Fan Dongrui[1,2], and Tang Zhimin[1,2]
[1](*State Key Laboratory of Computer Architecture* (*Institute of Computing Technology*, *Chinese Academy of Sciences*), *Beijing* 100190)
[2](*University of Chinese Academy of Sciences*, *Beijing* 100049)

Abstract Recently, the emerging graph neural networks (GNNs) have received extensive attention from academia and industry due to the powerful graph learning and reasoning capabilities, and are considered to be the core force that promotes the field of artificial intelligence into the "cognitive intelligence" stage. Since GNNs integrate the execution process of both traditional graph processing and neural network, a hybrid execution pattern naturally exists, which makes irregular and regular computation and memory access behaviors coexist. This execution pattern makes traditional processors and the existing graph processing and neural network acceleration architectures unable to cope with the two opposing execution behaviors at the same time, and cannot meet the acceleration requirements of GNNs. To solve the above problems, acceleration architectures tailored for GNNs continue to emerge. They customize computing hardware units and on-chip storage levels for GNNs, optimize computation and memory access behaviors, and have achieved acceleration effects well. Based on the challenges faced by the GNN acceleration architectures in the design process, this paper systematically analyzes and introduces the overall structure design and the key optimization technologies in this field from computation, on-chip memory access, off-chip memory access respectively. Finally, the future direction of GNN acceleration structure design is prospected from different angles, and it is expected to bring certain inspiration to researchers in this field.

Key words graph neural network; hybrid execution pattern; acceleration architecture; artificial intelligence; domain-specific architecture

摘 要 近年来,新兴的图神经网络因其强大的图学习和推理能力,得到学术界和工业界的广泛关注,被认为是推动人工智能领域迈入"认知智能"阶段的核心力量.图神经网络融合传统图计算和神经网络的

收稿日期:2021-03-01;**修回日期**:2021-04-26
基金项目:国家自然科学基金项目(61732018,61872335,61802367);中国科学院国际伙伴计划(171111KYSB20200002);数学工程与先进计算国家重点实验室开放基金(2019A07)
This work was supported by the National Natural Science Foundation of China (61732018, 61872335, 61802367), the International Partnership Program of Chinese Academy of Sciences(171111KYSB20200002), and the Open Project Program of the State Key Laboratory of Mathematical Engineering and Advanced Computing (2019A07).
通信作者:严明玉(yanmingyu@ict.ac.cn)

执行过程,形成了不规则与规则的计算和访存行为共存的混合执行模式.传统处理器结构设计以及面向图计算和神经网络的加速结构不能同时应对2种对立的执行行为,无法满足图神经网络的加速需求.为解决上述问题,面向图神经网络应用的专用加速结构不断涌现,它们为图神经网络定制计算硬件单元和片上存储层次,优化计算和访存行为,取得了良好的加速效果.以图神经网络执行行为带来的加速结构设计挑战为出发点,从整体结构设计以及计算、片上访存、片外访存层次对该领域的关键优化技术进行详实而系统地分析与介绍.最后还从不同角度对图神经网络加速结构设计的未来方向进行了展望,期望能为该领域的研究人员带来一定的启发.

关键词　图神经网络;混合执行模式;加速结构;人工智能;领域专用架构

中图法分类号　TP303

人工智能时代,包括卷积神经网络(convolutional neural networks, CNNs)、循环神经网络(recurrent neural networks, RNNs)等在内的机器学习应用为社会与生活的智能化做出了革新性的巨大贡献.然而传统的神经网络只能处理来自欧几里得空间(Euclidean space)的数据[1],该类分布规整且结构固定的数据无法灵活地表示事物间的复杂关系.现实生活中,越来越多的场景采用图作为表征数据属性与关系的结构.非欧几里得空间中的图结构理论上能够表征世间万物的互联关系(如社交网络、路线图、基因结构等)[2],具有极为丰富和强大的数据表达能力.图计算是一种能够对图进行处理,深入挖掘图数据内潜藏信息的重要应用,但其不具备对图数据进行学习的能力.

受到传统神经网络与图计算应用的双重启发,图神经网络(graph neural networks, GNNs)应运而生.图神经网络使得机器学习能够应用于非欧几里得空间的图结构中,具备对图进行学习的能力.目前图神经网络已经广泛应用到节点分类[3]、风控评估[4]、推荐系统[5]等众多场景中.并且图神经网络被认为是推动人工智能从"感知智能"阶段迈入"认知智能"阶段的核心要素[6-8],具有极高的研究和应用价值.

图神经网络的执行过程混合了传统图计算和神经网络应用的不同特点.图神经网络通常包含图聚合和图更新2个主要阶段.1)图聚合阶段的执行行为与传统图计算相似,需要对邻居分布高度不规则的图进行遍历,为每个节点进行邻居信息的聚合,因此这一阶段具有极为不规则的计算和访存行为特点.2)图更新阶段的执行行为与传统神经网络相似,通过多层感知机(multi-layer perceptrons, MLPs)等方式来进行节点特征向量的变换与更新,这一阶段具有规则的计算和访存行为特点.

图神经网络的混合执行行为给应用的加速带来极大挑战,规则与不规则的计算与访存模式共存使得传统处理器结构设计无法对其进行高效处理.图聚合阶段高度不规则的执行行为使得CPU无法从其多层次缓存结构与数据预取机制中获益.主要面向密集规则型计算的GPU平台也因图聚合阶段图遍历的不规则性、图更新阶段参数共享导致的昂贵数据复制和线程同步开销等因素无法高效执行图神经网络[9].而已有的面向传统图计算应用和神经网络应用的专用加速结构均只关注于单类应用,无法满足具有混合应用特征的图神经网络加速需求.因此为图神经网络专门设计相应的加速结构势在必行.

自2020年全球首款面向图神经网络应用的专用加速结构HyGCN[9]发表后,短时间内学术界已在该领域有多篇不同的硬件加速结构成果产出.为使读者和相关领域研究人员能够清晰地了解图神经网络加速结构的现有工作,本文首先对图神经网络应用的基础知识、常见算法、应用场景、编程模型以及主流的基于通用平台的框架与扩展库等进行介绍.然后以图神经网络执行行为带来的加速结构设计挑战为出发点,从整体结构设计以及计算、片上访存、片外访存多个层次对该领域的关键优化技术进行详实而系统的分析与介绍.最后还从不同角度对图神经网络加速结构设计的未来方向进行了展望,期望能为该领域的研究人员带来一定的启发.

当前已有的图神经网络应用领域综述论文从不同角度对图神经网络算法以及软件框架进行总结与分析.综述[1]对应用于数据挖掘和机器学习领域的主流图神经网络算法进行分类,并讨论不同类别算法的关系与异同.综述[10]依据图神经网络模型的结构和训练策略的不同,提出新的分类方法,并以模型的发展历史为主线进行介绍与分析.综述[11]围绕图的表示学习(representation learning)方法展开,并

建立统一的框架来描述这些相关模型.综述[12]关注于图神经网络的理论属性,总结图神经网络的表达能力(expressive power)并对比分析克服表达限制的图神经网络模型.综述[13]基于计算机的金字塔组织结构,对面向图计算的加速结构进行分类和总结,对于新兴的图神经网络应用,仅以 HyGCN[9]作为案例进行了讨论.与前述工作侧重点不同的是,本文针对图神经网络加速结构设计过程中涉及到的关键优化技术,进行系统性分析和总结,具有重要意义与启发价值.

1 图与图神经网络

本节首先对图数据结构和图神经网络的基础知识进行简要描述,然后对主流的图神经网络算法、图神经网络的应用场景以及图神经网络与传统应用的异同比较进行介绍.

1.1 图数据结构

图是一种由节点和边组成的数据结构,节点通过边进行连接来表征节点之间的关系.通常将图以 $G=(V,E)$ 的方式进行定义,其中 V 代表节点集合,E 代表节点之间的边集合.$v_i \in V$ 代表编号为 i 的节点,$e_{ij}=(v_i,v_j) \in E$ 代表从 i 号节点到 j 号节点相连的一条边.每个源节点与不同的目的节点相连接形成源节点的邻居集合,也即 $N(v)=\{u \in V \mid (v,u) \in E\}$.另外,节点 v 通过特征向量 $\boldsymbol{h}_v$ 来表征自己的属性.不同于传统图计算中仅用一个元素表征节点属性,图神经网络中节点的特征向量通常包含成百上千个元素,该规模被称为特征向量维度(feature dimension).本文用 $\boldsymbol{h}_v$ 表示节点 v 的特征向量,$|\boldsymbol{h}_v|$ 表示节点 v 的特征向量维度.例如对于包含 n 个元素的特征向量 $\boldsymbol{h}_v=(x_1,x_2,\cdots,x_n)$,其特征向量维度为 n.

邻接矩阵(adjacency matrix)是最直观与最简单的表示图中节点分布与相连关系的方式.对于具有 n 个节点的图 G,可将 G 中节点的相连关系表示为规模为 $n \times n$ 的矩阵 $\boldsymbol{A}$ 的形式.当节点 i 与节点 j 之间有相连边($e_{ij} \in E$)时,其在矩阵 $\boldsymbol{A}$ 中的对应位置元素 A_{ij} 置为 1,否则置为 0.使用这种存储方式,邻接矩阵只能表征图中节点的相连关系,而每个节点的属性(特征向量)需要另外存储在一个多维矩阵 $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ 中,其中 n 为节点个数,$d=|\boldsymbol{h}_v|$($\boldsymbol{h}_v \in \mathbb{R}^d$)为节点特征向量维度[1].

由于图结构的不规则性,其邻接矩阵通常为稀疏矩阵.有 3 种主流格式常用以存储稀疏图邻接矩阵:坐标列表格式(coordinate list, COO)、压缩稀疏行格式(compressed sparse row, CSR)以及压缩稀疏列格式(compressed sparse column, CSC).COO 是最简单的一种存储稀疏矩阵的方式,它通过 3 个一一对应的数组记录矩阵中所有非零的元素,3 个数组分别记录非零元素在矩阵中的行号、列号及节点特征.COO 的方式简单直观,但记录信息较多存在冗余,CSR 和 CSC 格式进一步对矩阵的存储进行压缩.CSR 格式同样通过 3 个数组对稀疏矩阵进行存储,但对行数组进行了压缩,具体方法是行数组中的每个元素依次记录稀疏矩阵中每行第 1 个非零元素在列数组中的偏移位置,因此也被称为偏移数组.列数组依次记录对应行的非零元素列号,也即单跳(1-hop)邻居(目标节点)的编号,该数组也被称为边数组,用以存储出边.最后通过属性数组记录节点的特征.CSC 格式与 CSR 形式相似,不同的是进行列的压缩,其边数组用以存储入边[9].

表 1 列出了本文所涉及到的标识及含义.

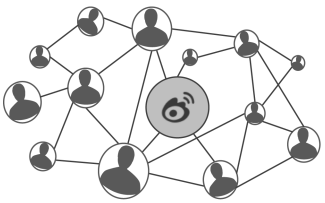
Table 1 Commonly Used Notations and Meanings in GNNs
表 1 图神经网络中常用标识及含义

标识	含义
G	图 $G=(V, E)$
V	图 G 的节点集合
E	图 G 的边集合
v_i	编号为 i 的节点
e_{ij}	节点 i 和节点 j 之间的边
$\boldsymbol{A}(A_{ij})$	邻接矩阵(矩阵中的元素)
$\boldsymbol{h}_v$	节点 v 的特征向量
$\boldsymbol{h}'_v$	节点 v 的聚合中间特征
$\lvert\boldsymbol{h}_v\rvert$	节点 v 的特征向量维度
$\boldsymbol{X}$	特征矩阵
$N(v)(S(v))$	V 的邻居集合(采样子集)
$\boldsymbol{W}$	权重矩阵
b	偏置量

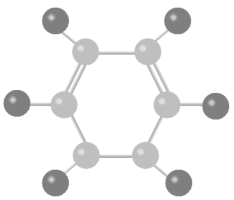
图结构具有极为强大的表示能力,理论上能够表征任何事物间的互联关系,在我们的日常生活中无处不在.例如图 1(a)是常见的城市地铁线路图,其中节点代表每个地铁站点,边代表站点之间的通行路径;图 1(b)是微博社交网络图,其中节点代表微博用户,边代表用户之间的关注与好友关系;图 1(c)是化学中的分子结构图,其中节点代表组成分子的原子,边代表原子之间的化学键.图结构数据中蕴藏

着丰富的信息,因此具有极高的实用和研究价值.图神经网络是一种典型的处理图结构数据并深入挖掘潜藏信息的应用,其应用场景和重要性将在1.4节中加以介绍.

(a) 地铁路线图



(b) 社交网络图



(c) 分子图

Fig. 1　Typical graphs in daily life

图1　生活中的典型图结构数据

现实世界中的图具有3个显著特征:

1) 规模大.实际应用场景中的真实图规模往往非常大,据统计,阿里巴巴的用户产品图在2018年时达到10亿用户与20亿产品的规模[14],脸书(Facebook)的数据图在2015年时包含超过20亿节点表示用户以及超过万亿条边表示用户间的好友、点赞等不同的关系[15],拼趣(Pinterest)网站在2018年已达到超过20亿节点和170亿条边的规模[16].除边和节点规模外,用于表征现实图中节点和边属性的特征向量维度常常也数以千计,进一步扩大了图的规模.

2) 幂律分布.现实图中的节点往往呈现幂律分布(power-law)的特征,即少数节点会与大部分节点之间具有相连关系,而大多数节点仅与少量其他节点之间存在边相连.例如,在电商的交易行为图中,少数的大型企业及销量高的商户节点会与极高数目的用户节点之间存在交易行为,而图中占大比例的普通买家用户节点仅与少量购买过商品的商户节点有边相连.

3) 动态多样性.现实场景中的图往往具有动态变化且邻居节点分布不规则的特征.例如,在微博等社交平台网络中,注册用户的数目、用户的关注数、点赞评论行为等每时每刻都在发生变化,且图中用户节点间的关系极为不规则.另外,现实图的种类多变,在不同的应用场景中,根据不同节点之间是否有指向性要求,分为有向图或无向图;根据不同节点的类型是否相同,分为同质图和异质图;根据处理过程中图结构是否发生变化,分为动态图和静态图.

1.2　图神经网络模型

尽管传统的神经网络在人工智能领域取得了巨大成功,但它只能应用于欧几里得空间的分布规整且结构固定的数据.为使更多现实应用场景智能化,图神经网络应运而生.图神经网络同时受到传统图计算和神经网络的启发,扩宽和加深了神经网络的应用范围和学习能力.图神经网络利用图结构,对节点的属性与相连关系进行建模与学习,通过对输入的节点、边、特征属性等信息进行逐层的迭代处理,最终对指定节点的特征向量进行更新,实现分类、预测、推荐、识别等不同的执行目的.

现有的主流图神经网络算法可以统一抽象为模型:

$$\boldsymbol{h}'_v=Aggregate(\boldsymbol{h}_{u\in N(v)}^{(k-1)}),$$

$$\boldsymbol{h}_v^{(k)}=Update(\boldsymbol{h}_v^{(k-1)},\boldsymbol{h}'_v),$$

其中$\boldsymbol{h}_v^{(k)}$表示节点v在第k层中更新得到的特征向量.整体而言,图神经网络是传统图计算和神经网络的混合体.在每层的执行过程中,图神经网络首先遍历整张图(或采样后的子图),对每个节点进行邻居信息的聚合(aggregate)获得该节点在本层的中间特征$\boldsymbol{h}'_v$,该过程与传统图计算相似;其后对聚合了邻居信息的中间特征与节点在上一层得到的特征进行组合,更新(update)获得本层该节点的输出特征向量$\boldsymbol{h}_v^{(k)}$,该过程与传统神经网络相似.

另外,为了简化执行过程,图神经网络算法通常可以在图聚合阶段增加自环(self-loop),也就是将节点自身的特征与其邻居节点特征同时进行聚合,进而在图更新阶段不区分自身节点特征和由邻居节点聚合形成的中间特征的神经网络参数.该方法尽管会损失部分模型表达能力,但执行过程简单并可从一定程度上缓解图神经网络可能存在的过拟合(over-fitting)现象[17].增加自环的图神经网络可抽象为模型:

$$\boldsymbol{h}'_v=Aggregate(\boldsymbol{h}_v^{(k-1)},\boldsymbol{h}_{u\in N(v)}^{(k-1)}),$$

$$\boldsymbol{h}_v^{(k)}=Update(\boldsymbol{h}'_v).$$

与神经网络类似,为获得对知识进行学习和预测的能力,一个完整的图神经网络包含训练(train)和推断(inference)两个主要部分.

1) 训练过程

训练是一种知识学习的过程,通过对网络进行

训练不断修正模型中的相应参数,才能为图神经网络应用提供更准确与强大的推断能力.

图神经网络通常可以通过反向传播(back propagation)的过程对损失函数进行训练[18-20].其中损失函数用于衡量图神经网络中最终得到的节点预测值与其真实值的差距,训练的目的是通过不断降低损失函数梯度以期模型对数据的预测更为准确,该过程通常可以采用随机梯度下降或其变形方法实现[17,21].

依据学习任务的标签信息,图神经网络的训练过程可分为有监督学习(supervised learning)、半监督学习(semi-supervised learning)和非监督学习(unsupervised learning).有监督学习的学习样本包含所有数据标签,通常用于整张图级别的分类,预测其类别标签[22-25];半监督学习的学习样本仅包含部分已知的数据标签,而图中其余节点的标签未知,通常用于节点级别的分类预测[3];非监督学习的学习样本不进行专门的数据标记,通过边等图中信息学习图的表示和发现潜在结构特征[26-28].

另外,许多已有的图神经网络算法为提高训练的效率提出了不同策略[11,17].以 GraphSAGE[28] 为代表的众多图神经网络算法提出采样训练的策略,不同于传统需要用整个图的拉普拉斯矩阵(Laplacian matrix)[3]作为训练样本,这些算法通过不同的采样策略对图的子集进行训练,从而减少冗余计算并可实现归纳式学习(inductive learning)的效果[16,29-30].Veličković等人[31], GPT-GNN[32]等工作提出通过预训练的方式,在没有或极少有标签数据的情况下,捕获和学习图的结构和语义信息,从而实现仅通过少量微调即可高效预测同领域图的效果.

2) 推断过程

通过训练过程对知识进行学习并不断对网络模型进行调整后,图神经网络具备了一定的推断能力,此时可执行推断过程,针对不同需求,对新知识进行推断.

推断同样遵循本节前述的通用图神经网络模型.此过程主要包含图聚合(aggregate)和图更新(update)两大主要阶段,不同的图神经网络算法可以通过不同的方法分别对不同阶段进行实现.图聚合阶段对每个节点的邻居进行遍历,并收集邻居节点的信息从而聚合为该节点在当前层的中间特征.图聚合阶段的行为通常可以由累加(accumulate)、均值(mean)、最大(max)、最小(min)、池化(pooling)等方法实现.图更新阶段对聚合过的节点特征向量通过神经网络来进行节点的变换与更新,为节点生成新的特征向量.图更新阶段的行为通常可以由多层感知机(multi-layer perceptron, MLP)、门控循环单元(gated recurrent unit, GRU)、长短期记忆网络(long short-term memory, LSTM)以及非线性激活函数等实现,而图更新阶段所需的权重、偏差等模型参数均通过训练阶段学习得到.另外还可通过读出(readout)操作,将获取的图中各节点的输出特征整合为整张图的特征表示.

为提升模型的执行性能与效果,主流的图神经网络算法也为图聚合及图更新阶段提出了不同的优化执行策略.例如,GCN[3]为图聚合阶段增加归一化(normalization)操作,从而削弱不同节点的度数差异引起的梯度不稳定等现象;GraphSAGE[28]、He等人[33]工作在图更新阶段引入向量串连(vector concatenation)、跳跃连接(skip connection)等操作,缓解网络层数不断加深后可能引起的节点表示过平滑(over-smoothing)的问题.

1.3 典型图神经网络算法

随着图神经网络的兴起,大量图神经网络算法和模型被提出,用于不同的应用场景,实现不同的学习与推理目的.如图 2 所示,常见的图神经网络应用可以分为卷积图神经网络(convolutional graph neural networks)、循环图神经网络(recurrent graph neural networks)、图自编码器(graph autoencoders)和时空图神经网络(spatial-temporal graph neural networks)四大类[1,34],本节将依次对这四大类图神经网络进行简要介绍.

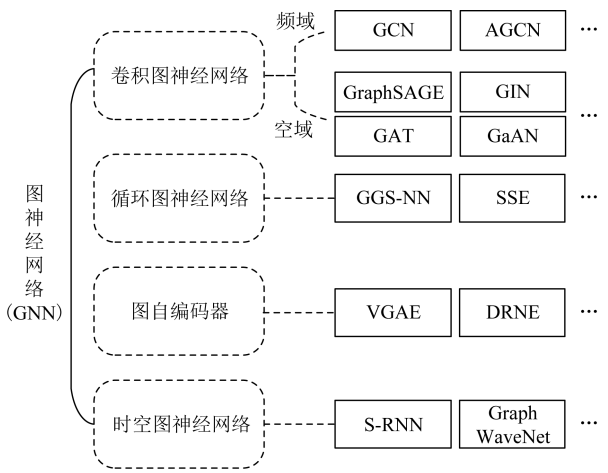


Fig. 2 Classification of mainstream graph neural network algorithms

图 2 主流图神经网络算法分类

1) 卷积图神经网络

卷积图神经网络(convolutional graph neural

networks)是目前研究和应用范围最广、相关变形算法种类最多的图神经网络分支之一.它将传统的卷积神经网络进行扩展,使其能够处理非欧几里得空间的图数据.卷积图神经网络还可分为频域(spectral)和空域(spatial)两种类型.频域卷积图神经网络通常利用拉普拉斯矩阵对图实现傅里叶变换(Fourier transform, FT)和逆傅里叶变换(inverse Fourier transform, IFT).空域卷积图神经网络利用图的信息传播来实现卷积,通常会比频域卷积图神经网络具备更强的灵活性和更高的执行效率.主流的卷积图神经网络算法包括图卷积网络(graph convolutional network, GCN)[3]、自适应图卷积神经网络(adaptive graph convolutional neural network, AGCN)[35]、采样聚合图神经网络(graph sample and aggregate, Graph-SAGE)[28]、图同构网络(graph isomorphism network, GIN)[36]、图注意力网络(graph attention network, GAT)[37]、门控注意力网络(gated attention network, GaAN)[38]等.

2) 循环图神经网络

循环图神经网络(recurrent graph neural networks)将循环神经网络(recurrent neural networks, RNNs)中的门控机制引入到图领域,循环执行从而获取图中节点的高层表示.主流的循环图神经网络包括门控序列图神经网络(gated graph sequence neural networks, GGS-NNs)[39]和随机稳态嵌入(stochastic steady-state embedding, SSE)[4].GGS-NNs将门控循环单元(gated recurrent units, GRUs)[40]引入到图神经网络中,形成门控图神经网络(gated graph neural networks, GG-NNs),并按顺序执行多个GG-NNs从而输出序列化结果.SSE在执行过程中交替进行节点稳态计算与模型参数调优,是一种能够对图进行稳态学习的循环图神经网络,并能够很好地扩展到大规模图中.

3) 图自编码器

图自编码器(graph autoencoders)将自编码器引入图领域中,通过编码器将节点表示转换为低维向量.图自编码器广泛应用于无监督学习领域[41],进行图的节点表示学习或生成新图.常见的图自编码器有变分图自编码器(variational graph autoencoders, VGAE)[26]、深度递归网络嵌入(deep recursive network embedding, DRNE)[42]等.VGAE是第1个将变分自编码器(variational autoencoder, VAE)引入图领域的算法.VGAE采用GCN作为编码器,以及一个简单的内积作为解码器.DRNE将节点的邻居进行排序,然后直接利用归一化的LSTM,通过聚合邻居节点的信息,对低维度节点向量进行重建.

4) 时空图神经网络

时空图神经网络(spatial-temporal graph neural networks)的核心目的是同时获取时空图(spatial-temporal graph)中时间和空间的依赖关系.时空图中每个节点的输入动态变化.时空图神经网络的目标为预测节点或时空图的数据标签[1],在天气预测、交通预测、动作识别等领域有广泛应用.主流的时空图神经网络包括结构神经网络(structural-RNN, S-RNN)[43]、图波网(Graph WaveNet)[44]等.S-RNN将循环神经网络RNN和图时空建模相结合,提出了一个周期性的框架来预测每个时间步的节点标签.Graph WaveNet在图卷积层通过有监督训练获取隐藏的空间依赖关系,同时通过堆叠扩展的因果卷积层(dilated casual convolutions)来获取隐藏的时间依赖关系,从而获取更高的时空网络执行效率和性能.

1.4　图神经网络应用场景与重要性

得益于对图结构数据强大的处理和学习能力,图神经网络在愈加广泛的现实应用场景中大放异彩,对为人们提供更加方便智能化的生活服务、为企业提供风险保障、提升业务质量和用户体验、推动科学技术快速发展等方面均具有重大意义.如图3所示,常见的图神经网络应用场景可分为推荐系统、计算机视觉、自然语言处理、自然科学研究、实况预测、金融风控六大类,本节将依次对不同的应用场景进行介绍.

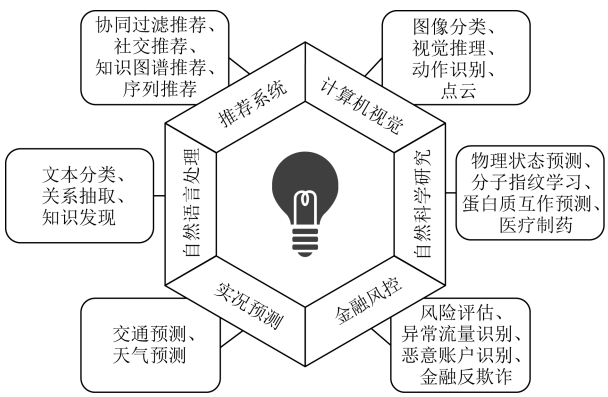


Fig. 3　Common application scenarios of graph neural networks

图3　常见的图神经网络应用场景

1) 推荐系统

推荐系统(recommendation system)是生活中

最常见的图神经网络应用场景之一,多用于社交、电商等具有大量用户交互的线上平台.推荐系统的图中包含用户节点和项目节点(例如商品、电影、歌曲等),通过分析不同节点间的交互关系与属性等附加信息,推荐系统为项目节点进行重要性排序,力求准确推断用户的偏好,为用户提供优质的内容推荐服务[45].

GC-MC[46],SpectralCF[47],NGCF[48]等工作利用协同过滤(collaborative filtering, CF)来处理矩阵补全问题,从而预测用户喜爱的项目排名.考虑到用户购买商品等行为会受到朋友的影响,DiffNet[49],DANSER[50],GraphRec[51]等工作将不同用户间的社交信息纳入用户节点与项目节点的关系分析中.KGCN[52],KGAT[53]等相关工作通过建立知识图谱,深入挖掘图节点之间的关联属性.另外还有DGRec[54],SR-GNN[55]等工作分析用户的过往行为,基于会话对用户的后继行为进行预测,从而实现序列推荐(sequential recommendation).

2) 计算机视觉

图神经网络在计算机视觉(computer vision)领域发挥重要作用.具体而言,主要应用于图像分类(image classification)、视觉推理(visual reasoning)、人体动作识别(action recognition)和点云建模等场景中.

图像分类是计算机视觉中最基础也极为重要的分支,文献[56-60]利用知识图谱、图像相似度分析、引入语义信息等方法对图像进行零样本(zero-shot)及少样本(few-shot)学习,从而高效地对图像进行分类.文献[61-64]通过建立场景图、知识图谱等方式处理计算机视觉中经典的视觉问答(visual question answering, VQA)任务,用于推理图像中的有效信息.文献[65-68]关注于人体关节之间的关联关系与时间序列,应用不同的图神经网络算法,实现基于人体骨架数据的动作识别(skeleton-based action recognition).文献[69-71]通过激光检测与测量方法(light detection and ranging, LiDAR)获取点云数据,并在其上实施图神经网络,从而实现点云的分类与分割.

3) 自然语言处理

图神经网络在自然语言处理(natural language processing)领域的应用是当前的热门研究方向之一,其中最主要的是文本分类(text classification),另外还在关系抽取(relation extraction)和知识发现(knowledge discovery)等方向进行了探索.

文献[72-75]将文本中的单词组织成为图结构,在其上实施不同的图神经网络,从而能够挖掘文本词句间的内在联系,根据不同的目标实现文本的分类.文献[76-79]通过图神经网络学习语句的关系知识,并通过关系抽取模型实现自然语言的关系推理.文献[80]对自然语言语句进行图神经网络处理,生成语义图(semantic graph),从而用于知识发现.

4) 自然科学研究

图神经网络在推动包括物理、化学、生物等多种自然科学研究的发展中起到重要作用.在物理学科方面,文献[81-83]提出不同的物理互作网络(interaction network),通过物体现态和关联关系等预测物体未来的状态.在化学和生物学领域,例如分子、蛋白质化合物等依据其结构能够方便地形成图结构数据.文献[84-86]将分子表示为原子图或联结树的形式,通过不同的图神经网络学习和预测分子指纹(molecular fingerprints).文献[87-88]通过图神经网络学习和预测蛋白质间的相互作用.对蛋白质等复杂结构关系的学习进一步也可应用于制药、疾病分类等领域.

5) 实况预测

近年来,交通和天气等领域也广泛利用图神经网络进行实况预测,多采用时空网络模型,同时对时间和空间信息进行采集和分析.交通方面,图神经网络通过对司机、行人等道路用户、道路传感器以及诸如交通法规、拥堵情况各类附加属性信息进行建模,形成运输系统[89],为用户的便利出行提供实时的道路预报.文献[90-93]根据交通流量预估出行速度等路况信息,文献[94]关注于网约车需求预测,从而优化城市资源分配.天气方面,文献[95]对栅格气候数据进行图神经网络建模,分析气候依赖,对厄尔尼诺现象(El Niño-Southern Oscillation, ENSO)等极端气象进行预测,从而避免可能的灾难发生.

6) 金融风控

金融行业存在大量的风控需求,是图神经网络应用最广泛的领域之一.文献[96]面向中小型企业建立企业互动关系网络,为供应链中的企业合作关系进行风险评估.文献[97]对用户交易关系建立动态图,通过图神经网络识别异常行为,对每笔交易进行风险评估.文献[5]针对用户行为建立图结构,通过图神经网络进行设备聚集、时间聚集等分析,识别恶意账户,从而规避恶意账户通过大量注册“僵尸账号”进行洗钱、欺诈等目的.

1.5　图神经网络与传统应用的比较

图神经网络整体上可以视为传统图计算应用与神经网络应用的结合体,其同时具有这2类应用的特征,本节对图神经网络与传统应用的比较进行介绍.

1)与图计算应用的比较

① 数据类型.图神经网络与图计算均针对非欧几里得空间的不规则图结构数据进行处理.

② 执行行为.图神经网络的图聚合阶段行为与传统图计算应用类似,均为通过逐跳(hop)遍历图中节点收集并聚合邻居信息.但是通常在图计算应用处理的图中,节点特征只包含单个元素,用于表征节点的层次、距离等属性.而在图神经网络应用处理的图中,节点特征需要通过向量进行表示.向量中的元素个数往往成百上千,且在执行过程中存在动态变化,能够表达非常丰富的属性信息.尽管图中节点分布极不规则,但得益于更高的特征向量维度,图神经网络相较图计算而言,具备较高的数据空间局部性.

③ 学习能力.图计算应用不具备知识学习的能力,执行过程较为简单,通常应用于路径规划、页面排序、网络分析等场景.而图神经网络集成神经网络的行为,具备知识学习和推断的能力,适用场景更为广泛.

2) 与神经网络应用的比较

① 数据类型.神经网络只能处理欧几里得空间的规整数据,而图神经网络面向的是现实生活中更广泛存在的非欧几里得空间的不规则且动态变化的图结构数据.

② 执行行为.图神经网络的图更新阶段与传统神经网络应用相似.但传统神经网络中,依赖信息仅能以节点的属性形式存在,而图神经网络能够通过聚合过程,将图节点间的依赖关系在图中进行传播.

③ 学习能力.二者均包含训练和推断的过程,具备知识学习的能力.

④ 可解释性.许多复杂的神经网络仅支持黑箱操作[98],无法有效提供可解释性.然而可解释性为智能算法的判定结果提供可靠证明,是保障其有效性的重要依据.图神经网络的执行模式为可解释性带来更多便利与可能,目前已有少量可解释性方法的成果[99-100]产出.

2　图神经网络编程模型与框架

算法1是一个图神经网络的通用编程模型[101],可适用于各类图神经网络算法.编程模型的输入包括图结构数据G、图中每个节点$v(v\in V)$的初始特征x_v以及图神经网络执行层数或所需收集的最大邻居跳数(hop)K.首先对图中每个节点进行特征$\boldsymbol{h}_v$初始化并对节点的邻居节点进行采样,其中采样过程为可选项.然后依次为图中每个节点收集并聚合其采样后邻居节点的特征,进而通过MLP等常见神经网络对图中节点进行更新,生成新的特征向量.图神经网络的聚合和更新过程交替进行,直到执行K次后,进行输出.图神经网络的输出可以是图中每个节点的最终特征向量,也可以是通过readout等操作获取的整张图的特征表示.

算法1. 图神经网络通用编程模型.

① for 每个节点 $v\in V$ do

② 　初始化 $\boldsymbol{h}_v^{(0)}\leftarrow x_v$;

③ 　$S(v)\leftarrow Sample(N(v))$;

④ end for

⑤ for $k\leftarrow 1$ to K do

⑥ 　for 每个节点 $v\in V$ do

⑦ 　　$\boldsymbol{h}'_v\leftarrow Aggregate(\boldsymbol{h}_{u\in S(v)}^{(k-1)})$;

⑧ 　　$\boldsymbol{h}_v^{(k)}\leftarrow Update(\boldsymbol{h}_v^{(k-1)},\boldsymbol{h}'_v,\boldsymbol{W},b)$;

⑨ 　end for

⑩ end for

作为近年来人工智能领域最热门的研究方向之一,图神经网络已产出大量不同的算法.众多企业及研究团队开展了基于通用平台的面向图神经网络的框架与扩展库的研发工作.由于图神经网络应用与传统神经网络应用在很多执行方式方面有异曲同工之处,诸多已有工作均基于发展成熟的神经网络框架,例如PyTorch和Tensorflow等进行扩展,形成支持图神经网络应用的新型框架.这些框架与扩展库支持多种不同的图神经网络算法,大多已开源,方便用户在其上灵活地构造图神经网络.下面将对主流的图神经网络框架与扩展库进行介绍.

1) PyG

PyTorch Geometric(PyG)[102]由多特蒙德工业大学研究团队提出,是目前最常用的通用图神经网络扩展库,它基于PyTorch框架扩展而成,同时支持在CPU和GPU平台上运行,已开源.除了常见的图结构数据处理方法外,PyG还提供对关系学习(relational learning)和3D数据处理等方法的支持.

PyG为用户提供通用的消息传递(message passing)接口,使用户能够在PyG上快速清晰地实现有关图神经网络算法的新研究想法.在此接口下,用户只需要分别定义message和update函数以及

选择节点聚合模式,即可完成一个图神经网络算法的构建,实现对节点进行邻居聚合以及对节点进行更新的操作.

2) DGL

Deep Graph Library(DGL)[103]是由亚马逊人工智能研究院与纽约大学合作开发的开源图神经网络扩展库.DGL基于多种已有的神经网络框架扩展实现,目前已支持Tensonflow,PyTorch和MXNet,并最小化用户跨平台迁移图神经网络模型时的工作量.

DGL将图神经网络计算过程抽象为用户可配置的消息传递单元,并提取图神经网络中的稀疏矩阵乘与消息传递机制间的联系,将计算操作整合为泛化的稀疏-稠密型矩阵乘(generalized sparse-dense matrix multiplication, g-SpMM)与泛化的采样稠密-稠密型矩阵乘(generalized sampled dense-dense matrix multiplication, g-SDDMM).另外,DGL还引入了不同类别的并行策略,通过对g-SpMM采用节点级并行,对g-SDDMM采用边级并行的方式,使其具备高执行速度和访存效率.

3) AliGraph

AliGraph[101]是阿里巴巴团队发布的面向大规模图神经网络的研发和应用设计的一款开源分布式框架,其已经在阿里巴巴集团完成商用落地.

AliGraph的系统分为算子(operator)、采样(sampling)和数据存储(storage)这3个层次.算子层将不同的图神经网络算法拆解为系统中的采样(sample)、聚合(aggregate)和组合(combine)三个算子,并进行优化计算;采样层集成多种不同的采样策略,使其能够快速准确地生成训练样本;数据存储层通过灵活的图划分、对图中不同属性进行分别存储以及缓存热点数据等策略来实现高效的数据组织和存储.

3 图神经网络加速的挑战

图神经网络的复杂执行过程为其有效加速带来了诸多困难,本节将首先对图神经网络的执行行为以及导致通用平台和面向传统应用的加速结构无法高效执行图神经网络的原因进行分析,进而给出图神经网络加速结构设计过程中遇到的挑战.

3.1 图神经网络执行行为分析

图神经网络是图计算与神经网络的结合体,其执行过程同时具有这2种传统应用的特点,整体而言具有混合的执行行为.图聚合和图更新阶段的执行行为的对比总结如表2所示.混合的执行行为给图神经网络的高效执行带来极大挑战.

1) 图聚合阶段.在此阶段,图神经网络对每个节点的邻居节点进行遍历,从而采集和聚合邻居节点的特征信息,与图计算应用相似.这一过程的执行在很大程度上依赖于图结构的分布特征.而现实世界的图往往具有极高的稀疏性,其邻接矩阵的稀疏度高达99%[104-105],且图中节点之间的连接分布不规则,每个节点的邻居节点的数量和位置均不固定.上述行为和特性导致图神经网络的图聚合阶段存在大量的动态计算与不规则访存,受内存约束.

2) 图更新阶段.在此阶段,图神经网络对图中每个节点进行特征向量的神经网络变换和更新,与传统神经网络应用相似.这一过程中,不同节点共享相同的神经网络结构,且每层中神经元之间的连接也相同,因此图更新阶段具有静态的计算和规则的访存,受计算约束.

Table 2 Execution Behaviors in GNNs[9]

表2 图神经网络执行行为总结[9]

执行行为	图聚合阶段	图更新阶段
访存模式	间接&不规则	直接&规则
数据复用率	低	高
计算模式	动态&不规则	静态&规则
计算强度	低	高
执行约束	内存	计算

3.2 通用平台的不足

依据文献[106]所述,不同的图神经网络模型在图聚合和图更新阶段具有相似的执行特性,因此本节以图卷积神经网络的量化数据为例说明通用平台加速的不足之处.

表3列出了COLLAB[107]数据集在CPU(Intel Xeon CPU E5-2680 v3)平台上执行GCN模型的结果,表4列出了Reddit[107]数据集在GPU(Nvidia GPU V100)平台执行GraphSAGE模型的结果,上述实验均通过PyTorch Geometric实现.从表3中可以发现,图聚合阶段中,每个操作都比图更新阶段从DRAM访问更多的数据,从而导致更高的DRAM访问能耗.源于每个节点邻居的高度随机性,在图聚合阶段中L2和L3缓存的每千条指令缺失次数(miss-predicts per kilo-instructions, MPKI)极高,同时也导致无法预测特征向量的访存地址.不规则访存使得传统通用处理器的数据预取机制失效[108].

图更新阶段中,每个操作仅需要从 DRAM 访问少量数据,这是因为矩阵向量乘的计算量很大且多层感知机的权重矩阵在节点之间广泛共享.然而在 CPU 上对共享数据的复制和线程之间的同步,执行时间开销最多可达 36%.对于 GPU 平台上,也有类似的结果(如表 4 所示).

Table 3　Execution Behaviors of GNNs on CPU[9]

表 3　图神经网络执行行为在 CPU 上的结果[9]

执行行为量化指标	图聚合阶段	图更新阶段
DRAM 访存量或操作/B	11.6	0.06
DRAM 访存能耗或操作/nJ	170	0.5
L2 Cache MPKI	11	1.5
L3 Cache MPKI	10	0.9
同步时间占比		36%

Table 4　Execution Behaviors of GNNs on GPU[109]

表 4　图神经网络执行行为在 GPU 上的结果[109]

执行行为量化指标	图聚合阶段	图更新阶段
DRAM 访存量或操作/B	2.35	0.01
计算单元利用率/%	50	90
每周期完成指令数(IPC)	1.78	2.49
L2 Cache 命中率/%	6.87	82.5

图神经网络的混合执行行为,导致通用平台均无法为图神经网络的执行提供专属且高效的算力.对于 CPU 平台来说,它们缺少计算资源,并且图聚合阶段的遍历不规则性会导致频繁的数据替换.对于 GPU 来说,其结构本质上是面向类似神经网络的具有规则执行行为且计算密集型的应用进行加速[110],缺少应对不规则性的能力.

3.3　传统应用加速结构的不足

通用平台下的图神经网络框架对图神经网络应用的加速空间有限,设计专用的加速结构就成为了一种顺应而生的趋势.由于为特定领域设计专用的加速结构可以为特定的应用量身定制计算单元和内存层次结构,因此其成为解决现有架构执行效率低下的有效且普遍的方案.

然而面向传统图计算和神经网络应用的加速结构无法高效地应对图神经网络.对于图计算加速结构而言,它们主要面向动态稀疏计算和不规则访存进行设计,且处理的图结构中节点特征属性很小,因此不具备加速图更新阶段密集计算的能力,并且无法有效挖掘图更新阶段的规则性和较好的空间局部性.对于神经网络加速结构而言,它们主要面向静态密集计算和规则访存进行设计,尽管已有一些工作将神经网络加速结构向稀疏矩阵运算进行扩展,但图神经网络所处理图的稀疏度至少比传统神经网络所处理的图高 2 倍[105].因此神经网络加速结构不具备应对不规则访存和不规则计算的能力,无法高效执行图神经网络的图聚合阶段.另外,图计算加速结构和神经网络加速结构均只能针对图神经网络中的单一阶段进行加速,无法融合 2 个阶段的执行[9],不足以应对图神经网络应用的加速需求.

3.4　图神经网络加速设计挑战

通用平台及面向传统图计算和神经网络应用的加速结构处理图神经网络效率低下的原因总结如图 4 所示.已有处理器结构的效率低下使得为图神经网络专门设计相应的加速结构势在必行.

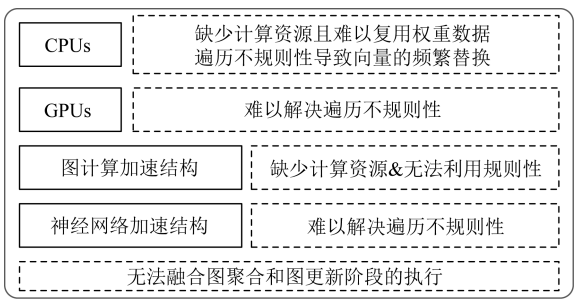


Fig. 4　Reasons for low efficiency in existing processors

图 4　现有处理器结构效率低下原因总结

然而由于图神经网络混合执行行为的特殊性,面向图神经网络应用的加速结构设计仍然面临诸多挑战,挑战主要可以分为计算和访存 2 个方面.计算方面:图神经网络加速结构需要同时能够高效应对不规则和密集规则型计算.由于图中节点具有极高的不规则性并服从幂律分布,图聚合阶段对邻居节点的遍历会导致严重的负载不均衡,为图神经网络加速结构的设计带来更多挑战.另外,图神经网络的执行过程中还潜藏着不同级别的并行性待加速结构挖掘.访存方面:图神经网络加速结构需要同时能够高效应对不规则和规则的粗粒度访存以及高带宽需求.同时如何充分地进行图数据复用也为图神经网络加速结构的设计提出更高要求.

4　图神经网络加速结构分类方案

自 2020 年全球首款面向图神经网络应用的专用加速结构 HyGCN[9] 发表后,短时间内学术界已在该领域产出了多篇不同的硬件加速成果.这些工作在所支持的图神经网络算法类别、所应用的阶段、

硬件加速平台、关键优化技术等方面有所差异,表5列出了现有工作的相关信息对比.

1) 支持算法方面

通过表5可以看到,目前现有工作大多针对单一种类图神经网络算法进行加速,尤其以图卷积神经网络GCNs居多.仅Auten等人,DeepBurning-GL,EnGN和Cambricon-G的研究工作能够支持多种类的图神经网络算法.

2) 支持阶段方面

现有工作大多只能支持对图神经网络的训练或推断中的单一阶段进行加速,仅有Cambricon-G同时支持2个阶段,GraphACT面向训练过程,其余工作均只关注推断阶段.

3) 加速平台方面

现有工作通常采用CPU-FPGA的异构平台、ASIC平台或存内计算平台实现硬件加速结构搭建.

FPGA是现场可编程逻辑门阵列(field programmable gate array),设计人员通过FPGA配置文件定义专用的逻辑电路,实现所需的硬件功能,具有较强的可配置性和灵活性.CPU-FPGA的异构平台,可以使CPU和FPGA平台各取所长,分工执行图神经网络应用中的不同步骤,并相互配合达到更好的效果.

ASIC为专用集成电路(application specific integrated circuit),用户能够根据不同的应用场景和用途定制专用的硬件结构.ASIC可为设计人员提供丰富的计算和存储资源.与FPGA相比,ASIC适合复杂度更高、规模更大的硬件结构设计.

存内计算(processing in memory, PIM)是解决硬件结构设计中存储墙(memory wall)问题的有力方案之一.存内计算通过修改内存相关的电路设计或引入新型存储等方式,在内存附近或在内存之中进行计算,能够有效拉近计算单元与所需数据间的距离,有助于为硬件结构提供更高的访存带宽.

4) 关键优化技术方面

针对第3节介绍的图神经网络执行行为带来的加速挑战,现有工作的关键优化技术可以归纳为计算、片上访存和片外访存这3个层次.

① 图神经网络加速结构在计算层次主要的优化目标是充分挖掘并行性,常见的优化方向包括负载均衡、脉动阵列、减少冗余计算与降低计算复杂度等.

② 目前片上访存层次主要的优化目标是深入挖掘粗粒度访存数据的空间局部性和时间局部性,尽可能减少片上数据的频繁替换.主要的解决思路是采用大容量片上存储和对图数据进行数据重排.另外还有新兴方法通过优化模型压缩图神经网络的模型参数数据,降低对片上存储空间的需求.

③ 现有的解决片外访存层次挑战的主要思路是基于特定的图数据划分方法提高预取效率和数据重用率,利用稀疏性消除、动态访存调度、数据结构重组提高带宽利用率,通过操作融合减少访存带宽需求等.

Table 5 Comparison Among Existing GNN Acceleration Architectures

表5 现有图神经网络加速结构对比

No	名称	支持算法	支持阶段	加速平台	关键优化技术		
					计算	片上访存	片外访存
1	HyGCN[9]	GCNs	推断	ASIC	√	√	√
2	Auten等人[105]	GCNs,GATs,MPNN,PGNN	推断	ASIC	○	√	○
3	GraphACT[111]	GCNs	训练	CPU-FPGA	√	√	√
4	Zhang等人[112]	GCNs	推断	CPU-FPGA	√	√	√
5	GCNAR[113]	DAGCN	推断	CPU-multi-FPGA	√	√	○
6	FPGAN[114]	GATs	推断	CPU-FPGA	√	√	√
7	DeepBurning-GL[115]	GCN,GS-Pool,R-GCN,EdgeConv	推断	FPGA	√	√	○
8	AWB-GCN[104]	GCNs	推断	ASIC	√	√	√
9	EnGN[116]	GCNs,GS-Pool,Gated-GCN,GRN,R-GCN	推断	ASIC	√	√	√
10	GNN-PIM[117]	GCN,CommNet,GGCN	推断	PIM	√	○	√
11	Cambricon-G[118]	GCN,GraphSAGE,DiffPool,EdgeConv,DGMG,GUN	训练、推断	ASIC	√	√	√

注:"√"表示包含该类关键优化技术;"○"表示未涉及该类关键优化技术.

大多数面向图神经网络应用的加速结构均从上述3方面进行了针对性优化,详细对比参照表5.由于关键优化技术是加速结构设计的核心内容,为使相关研究人员能够受到启发,本文将以不同层次的关键优化技术为分类标准,在接下来的2节对已有工作展开详细介绍.

5　图神经网络加速结构整体分析

本节对目前已有的图神经网络加速结构的设计核心进行总结,并从整体设计的角度,对加速结构如何融合不同层次的关键优化技术达到预期加速效果进行介绍。各层次的优化技术将在第6节中进行具体剖析。

由于图神经网络加速结构面临计算和访存2方面的挑战,因此高效的图神经网络加速结构通常都包含不同层次的优化思路和优化技术.并且,各个层次的关键优化技术之间相辅相成,甚至融为一体为同一个设计核心服务.现有图神经网络加速结构的整体设计核心可归纳为表6:

Table 6　Core Design of Existing GNN Acceleration Architectures

表6　现有图神经网络加速结构设计核心

类别	No.	名称	设计核心
混合加速结构	1	HyGCN[9]	混合结构设计高效执行GCNs
	2	Auten等人[105]	NoC连接定制的硬件单元以应对不规则访存
	3	GraphACT[111]	基于异构系统的混合结构设计加速GCNs训练
	4	Zhang等人[112]	算法-硬件结构协同优化
	5	GCNAR[113]	多FPGA异构平台上深度流水加速DAGCN图卷积神经网络模型
	6	FPGAN[114]	为GATs在FPGA平台下定制软硬件协同优化
	7	DeepBurning-GL[115]	基于定制优化的计算和访存模板为各种场景自动化生成图神经网络加速结构
一体化加速结构	1	AWB-GCN[104]	运行时自动调节负载均衡
	2	EnGN[116]	高吞吐高能效的一体化结构应对大规模图神经网络
	3	GNN-PIM[117]	面向图神经网络的存内计算加速结构
	4	Cambricon-G[118]	高效加速动态GNNs的训练和推断阶段

从整体设计核心出发,可将现有工作分为混合加速结构和一体化加速结构两大类.同时,这2类加速结构均配合不同层次的关键优化技术.各项关键优化技术能够应对图神经网络在各个层次面临的挑战,同时又相辅相成,共同实现图神经网络的高效执行.

5.1　混合加速结构

为了更有效地针对不同执行行为进行优化,混合加速结构将图神经网络算法解耦合为多个执行阶段,分别设计与不同阶段执行行为相匹配的计算单元和存储单元,从而有针对性地对不同阶段进行加速和提升执行效率.

如图5所示,HyGCN[9]分别为图聚合阶段和图更新阶段设计了Aggregation加速引擎和Combination加速引擎,并进行混合结构的协调控制.同时,HyGCN在计算层次分别对Aggregation和Combination加速引擎设计负载均衡策略以及引入脉动阵列,挖掘运算并行性并提升运算效率.在访存层次,HyGCN通过精巧设置的大容量片上缓存、滑动收缩窗口消除数据稀疏以及动态调度访存等策略提升片外访存效率与带宽利用率.访存层次的优化为计算单元的数据持续供给提供保障,从而有力支撑计算层次的优化技术更好地发挥作用.

GraphACT[111]、文献[112]和GCNAR[113]在CPU和FPGA的异构系统上执行图神经网络,使CPU和FPGA都能够分别对所擅长的执行行为进行加速.如图6所示,GraphACT在CPU上主要执行访存密集型或具有不规则执行行为的操作,例如Sample操作、去除冗余Aggregate操作的预处理、计算损失等;在FPGA上主要执行计算密集型操作,例如正向以及反向传输的Aggregate操作和Combine操作.同时GraphACT也在计算和访存层次上提出了多种优化技术.GraphACT在计算层次上,采用脉动阵列来挖掘运算并行性,提高运算效率;在访存层次上,通过设置不同的片上存储器尽可能多地存储计算所需的各类数据,提升数据复用率,

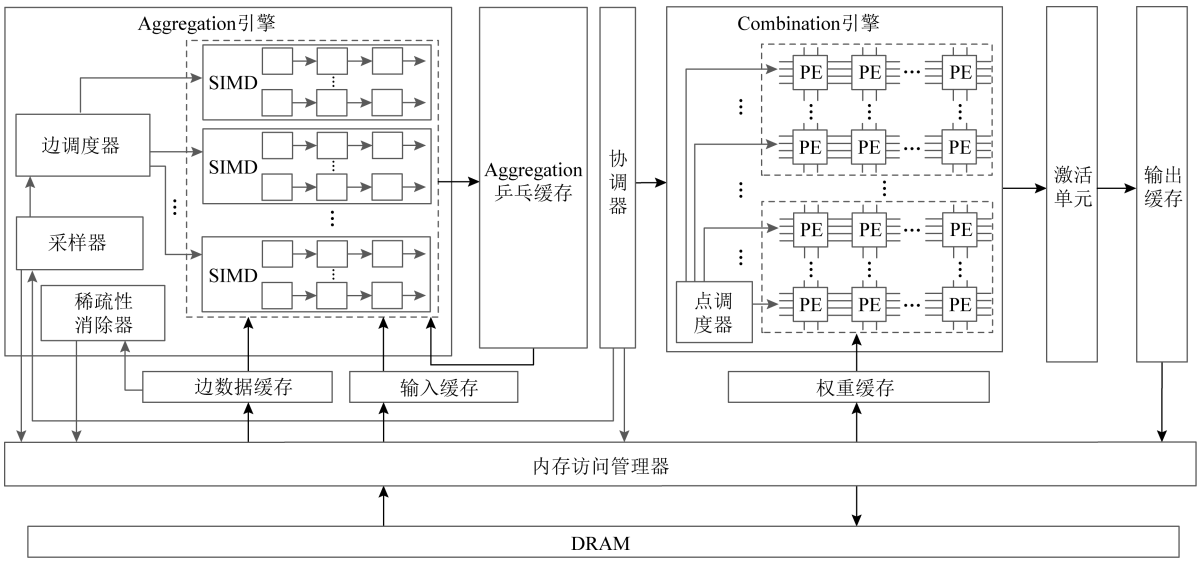


Fig. 5 Hybrid architecture of HyGCN[9]

图 5 HyGCN 的混合加速结构[9]

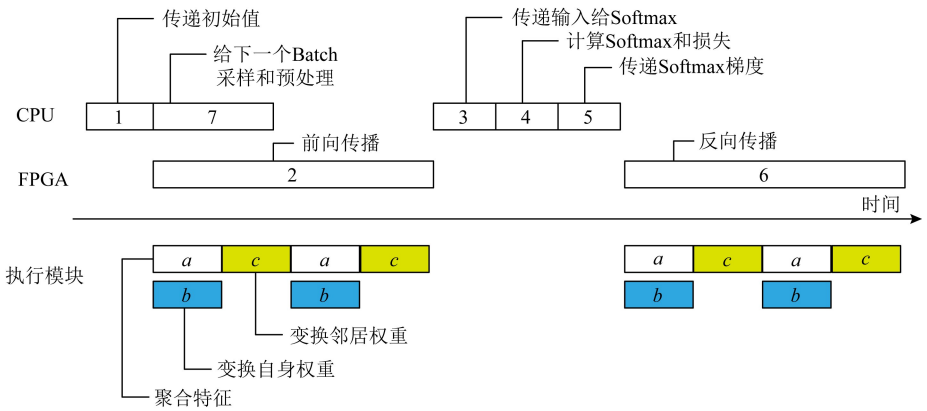


Fig. 6 Heterogenous pipeline of CPU-FPGA platform in GraphACT[111]

图 6 GraphACT 的 CPU-FPGA 异构流水[111]

使计算单元能够更快获取热点数据.另外,预处理过程中的稀疏性消除机制能够有效减少图中的冗余边,从而降低冗余的计算量和片外访存量,也即同时在计算和访存 2 个层次产生优化效果.

FPGAN[114] 通过软硬件协同优化的方式在 FPGA 平台上加速图注意力网络的执行.硬件方面,FPGAN 设计特征聚合模块和线性转换模块来分别执行图注意力网络的图聚合阶段和图更新阶段.软件方面,FPGA 通过转换图注意力网络的执行过程,在计算层次降低复杂度;通过压缩权重、操作融合的方式在访存层次降低对片上存储空间及片外带宽的需求.访存层次的优化配合复杂度降低的网络模型,能够为计算单元更快更多地提供操作数据.另外,软件方面的优化技术帮助图注意力网络适配定制的硬件处理模块,协同提升执行效率和性能.

5.2 一体化加速结构

与混合加速结构相对的,一体化加速结构对所有的计算资源进行统一调度,用于执行图神经网络的各个阶段,同时配合不同层次的关键优化技术,整体加速图神经网络的执行过程.

AWB-GCN[104] 以稀疏-密集矩阵乘(SpMM)的形式执行频域图卷积神经网络,并将 SpMM 的运算操作送入统一结构的 PE 中.同时,AWB-GCN 在计算层次设计运行时的负载调度机制以实现动态均衡负载,在访存层次辅以大规模片上存储器及图矩阵分块等方式提升数据局部性,减少片外访存带宽需求,为计算单元复用片上数据提供更优的支持.

EnGN[116] 加速结构中包含一组规模为 128×16

的同构处理单元(processing elements, PE)阵列.如图 7 所示,PE 之间以 Ring-Edge-Reduce(RER)的拓扑形式相连接,每一列的 PE 用于执行邻居聚合操作,而每一行的 PE 用于处理节点的特征属性.统一的加速结构之上,EnGN 在计算层次通过灵活选择图神经网络每层的阶段执行顺序减少冗余计算,在访存层次采用图分块及多层次片上存储器等方式降低片外访存需求,不同层次共同应对图神经网络执行行为所带来的不同挑战.

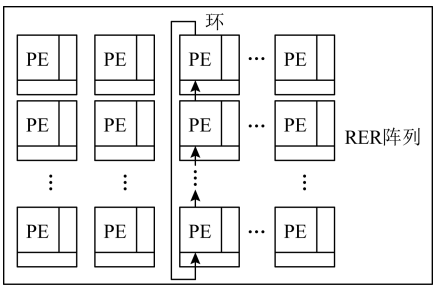


Fig. 7　Processing element array in EnGN[116]

图 7　EnGN 的处理单元阵列[116]

Cambricon-G[118]加速结构的运算由一个 3D 形式的立方体加速引擎(cuboid engine, CE)完成.CE 中包含多个同构的节点处理单元(vertex processing units, VPUs),不同的 VPU 组织为 2D 脉动阵列的形式,用于处理以邻接立方体形式存储的图数据.在计算层次,Cambricon-G 通过 VPU 组成的脉动阵列挖掘数据局部性和运算并行性;在访存层次,Cambricon-G 通过对邻接立方体进行划分,辅以混合的片上存储系统,实现多维度的数据重用,降低访存带宽需求.整体而言,Cambricon-G 通过访存层次的精细分块,将子块数据送入 VPU 阵列中进行运算,实现高效加速图神经网络的效果.另外,灵活的分块策略和拓扑感知 Cache 助力 Cambricon-G 有效应对动态变化的图.

6　图神经网络关键优化技术

本节将从计算、片上访存与片外访存这 3 方面对现有图神经网络加速结构的关键优化技术进行分类介绍.

6.1　计算优化层次

混合的执行行为给图神经网络加速结构在计算层次的结构设计与优化带来了巨大挑战.从关键优化技术角度看,图神经网络芯片在计算层次主要的优化目标是充分挖掘并行性和提高计算部件的利用效率,常见的优化方向包括负载均衡、脉动阵列、减少冗余计算及降低计算复杂度等,下面将依次对其进行介绍.

1) 负载均衡

由于图节点分布的不规则性,每个节点的邻居节点个数各不相同,使得在图神经网络的图聚合阶段中,每个节点的计算任务量差异巨大,导致严重的负载不均衡.如图 5 所示,HyGCN[9]设计 Aggregation 加速引擎,其中包含多个 SIMD(single instruction multiple data)计算单元和边调度器.边调度器会依据边表中每个节点对应的邻居节点信息,将邻居节点进行聚合,以特征向量中的元素为单位,分配到 SIMD 计算单元的不同通道(lane)中,从而均衡负载,并能够更加充分地开发边级并行性和节点内部的并行性.

如图 8 所示,GCNAR[113]提出的加速结构中包含多个 FPGA 节点,该工作将 DAGCN 图神经网络模型[119]中的注意力图卷积(attention graph convolution, AGC)层均衡分配到各个 FPGA 节点中,也即每个 FPGA 节点处理相同个数的图卷积层,从而保证每个 FPGA 节点的负载均衡.

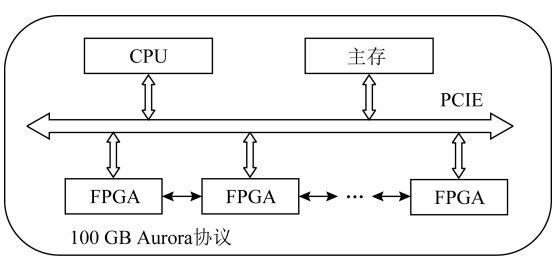


Fig. 8　CPU-multi-FPGA architecture of GCNAR[113]

图 8　GCNAR[113]的 CPU-多-FPGA 结构

AWB-GCN[104]基于稀疏矩阵设计了加速频域图卷积神经网络的加速结构,并利用运行时动态负载调度机制实现稀疏矩阵不均衡负载重新分配.运行时动态负载调度机制包括 3 种运行时工作负载重新平衡的技术:负载平滑分配(distribution smoothing)技术、负载远程交换(remote switching)技术以及矩阵行重映射(row remapping)技术.负载平滑分配技术在每轮计算过程中,实现邻居处理单元的负载均匀化.AWB-GCN 通过跟踪处理单元的任务队列(task queues, TQs)中的待处理任务数量,来监控运行时的计算单元利用率信息,进而将待处理任务量大的处理单元负载分担给较为空闲的邻居处理单元.负载远程交换技术通过在过载和较空闲处理单元(即利用率在波峰和波谷的处理单元)之间部分或

完全交换负载的方式解决负载局部集群问题(regional clustering).AWB-GCN 将包含过多非零元素(出入度极大)的行称为“邪恶行”(evil row),该行的负载无法通过上述 2 种技术在邻居间完美消化.矩阵行重映射技术在远程交换的基础上,将邪恶行负载进行重新映射.自动协调器负责判定是否需要进行行重映射,若需要,则将邪恶行的负载分配到最空闲的处理单元中,再通过邻居处理单元进行负载分担,从而进一步实现全局负载均衡.通过以上 3 种策略,AWB-GCN 连续监控稀疏矩阵实时运行信息,动态调整处理单元之间的负载分配,并在效果收敛后复用已取得的理想配置,解决 GCN 中大规模幂律分布结构数据导致的处理单元之间负载不均衡的问题.

GNN-PIM[117]加速结构提出了一种可重配的计算节点,可配置为对图节点进行处理的计算节点 vertex node 或对边进行处理的计算节点 edge node.GNN-PIM 根据图的拓扑分布情况,灵活地对计算节点进行配置,在一定程度上均衡负载.

2) 脉动阵列

对于图神经网络的图更新阶段的规则执行行为,受到 TPU[120]设计的启发,目前已有的图神经网络加速结构常采用脉动阵列来执行更新操作.

HyGCN[9]的 Combination 加速引擎包含多个由脉动阵列组成的脉动模块,脉动模块可以在独立执行和合作执行这 2 种模式下工作.在独立执行模式中,每个脉动模块独立完成一小组节点的矩阵向量乘(matrix-vector multiplication, MVM)操作.脉动阵列的一行用来完成同一节点特征向量的神经网络变换操作,每个脉冲模块内部的脉动阵列共享权重数据.在合作执行模式中,Combination 加速引擎中的所有脉动模块联合工作,对更多数量的节点同时进行神经网络变换操作,并共享权重数据.这 2 种模式都能够有效挖掘图更新阶段中的 MVM 并行性以及节点间的运算并行性,同时还能够提升权重数据的重用率.

与 HyGCN 类似,GraphACT[111]、文献[112]、GCNAR[113]、DeepBurning-GL[115]及 Cambricon-G[118]也同样采用脉动阵列在图神经网络的图更新阶段,对特征和权重进行矩阵运算,有效挖掘运算并行性,提高运算效率.

3) 减少冗余计算

GraphACT[111]通过预处理的方式,提前完成图中复用率较高的部分运算,从而去除邻居节点的冗余归约操作,能够有效减少计算量以及 CPU 与 FPGA 之间的通信开销.文献[112]同样采用预处理的方式,通过图稀疏化操作,合并公共邻居节点,消除高出/入度节点的边计算,其策略与 GraphACT 异曲同工.其不同点在于,GraphACT 应用于图神经网络的训练过程,文献[112]主要应用于推断过程,由于推断过程中整个图的冗余会比训练过程的 mini-batch 子图更多,因此该策略在文献[112]中的冗余消除效果和计算优化水平会更高.GCNAR[113]通过预处理,消除图邻接矩阵中全零的行,并重新组织邻接矩阵,从而减少算法的计算量.

另外,文献[112]和 EnGN[116]分析图神经网络的编程模型,同时为每一图神经网络层实现 2 种计算顺序,并能够根据前后层的特征维度判断采用哪种执行模式,从而有效降低计算量.第 1 种计算顺序是先执行图聚合阶段后执行图更新阶段,第 2 种计算顺序是先执行图更新阶段后执行图聚合阶段.

4) 降低计算复杂度

FPGAN[114]在不损失精确度的情况下,通过简化模型的方法,降低图注意力网络的计算复杂度,减少对 FPGA 中数字信号处理器(digital signal processors, DSPs)计算资源的需求,从而提升整体运算性能和效能.具体而言,首先,FPGAN 将原模型中的激活函数 leakyrelu 简化为 relu.其次通过特征量化(feature quantification)步骤 ,将每层的输入特征和激活因子(activation)从浮点域转入定点域.然后,FPGAN 通过直接检索查找表(lookup table, LUTs)的方式近似模拟 SoftMax 函数中的 exp 操作.经过以上步骤,FPGAN 可将图注意力模型中的所有浮点乘法操作转换为移位操作,有效降低了计算复杂度.另外,FPGAN 通过引入膨胀系数(expansion coefficient)等方法保证模型的修改不会影响最终的推断精确度.在该优化方法的帮助下,单个计算单元仅需要少量硬件资源即可实现,进而削弱图注意力模型的执行性能对 DSP 的依赖,提高 FPGA 的资源利用率,提升模型执行速度.

6.2 片上访存优化层次

图神经网络中节点的特征属性为高维数据,因此对节点属性的访问为粗粒度的不规则访问,这也是图神经网络与传统图计算执行过程的典型区别之一.不规则粗粒度访存导致的片上存储 Cache 命中率低的问题,为图神经网络加速结构的设计在片上存储层次带来极大挑战.解决这一挑战的主要思路分为 2 种:1)采用大容量的片上存储,并配合批量

处理与数据分割等技术;2)对图数据进行数据重排.这2种解决思路的目的是一致的,均为深入挖掘粗粒度访存数据的空间局部性和时间局部性,尽可能减少片上数据的频繁替换.另外还有新兴方法通过优化图神经网络模型、压缩权重等模型参数数据,降低对片上存储空间的需求.下面将依次对上述方法进行介绍.

1) 大容量片上存储

选取大容量的片上存储是最为常见的片上存储层次优化策略.HyGCN[9]、文献[105]、GraphACT[111]、AWB-GCN[104]、文献[112]、EnGN[116]、文献[114]、DeepBurning-GL[115]及Cambricon-G[118]等均采用该方法,并配合相应的批量处理与数据分割等技术,降低图神经网络中被不规则粗粒度访存的数据在片上和片外间频繁替换的次数.

如图5所示,HyGCN[9]为了挖掘图神经网络中涉及的各种数据的局部性,基于eDRAM(embedded DRAM)为具有不同访存模式的数据设计了不同的缓存(buffer).HyGCN通过边buffer来存储Aggregation加速引擎所需的边数据,挖掘边表的空间局部性;通过输入buffer来存储每层节点的输入特征向量;通过权重buffer来存储图更新阶段所需的权重矩阵,从而挖掘其时间局部性;通过输出buffer来合并每层待写回的节点特征向量.上述缓存均利用双缓冲(double buffer)技术来掩盖访存延迟.另外,HyGCN还在Aggregation和Combination加速引擎之间添加Aggregation缓存,用于存储图聚合阶段的中间数据,Aggregation缓存分为2个部分来实现乒乓缓冲(ping-pong buffer)机制,从而提升加速引擎之间的并行度.

文献[105]分别在其加速结构的不同模块中添加片上存储来缓存不同类型的数据,从而挖掘数据局部性.如图9所示,具体实施方案为:图处理单元(graph PE, GPE)中设置片上存储器来保存应用执行

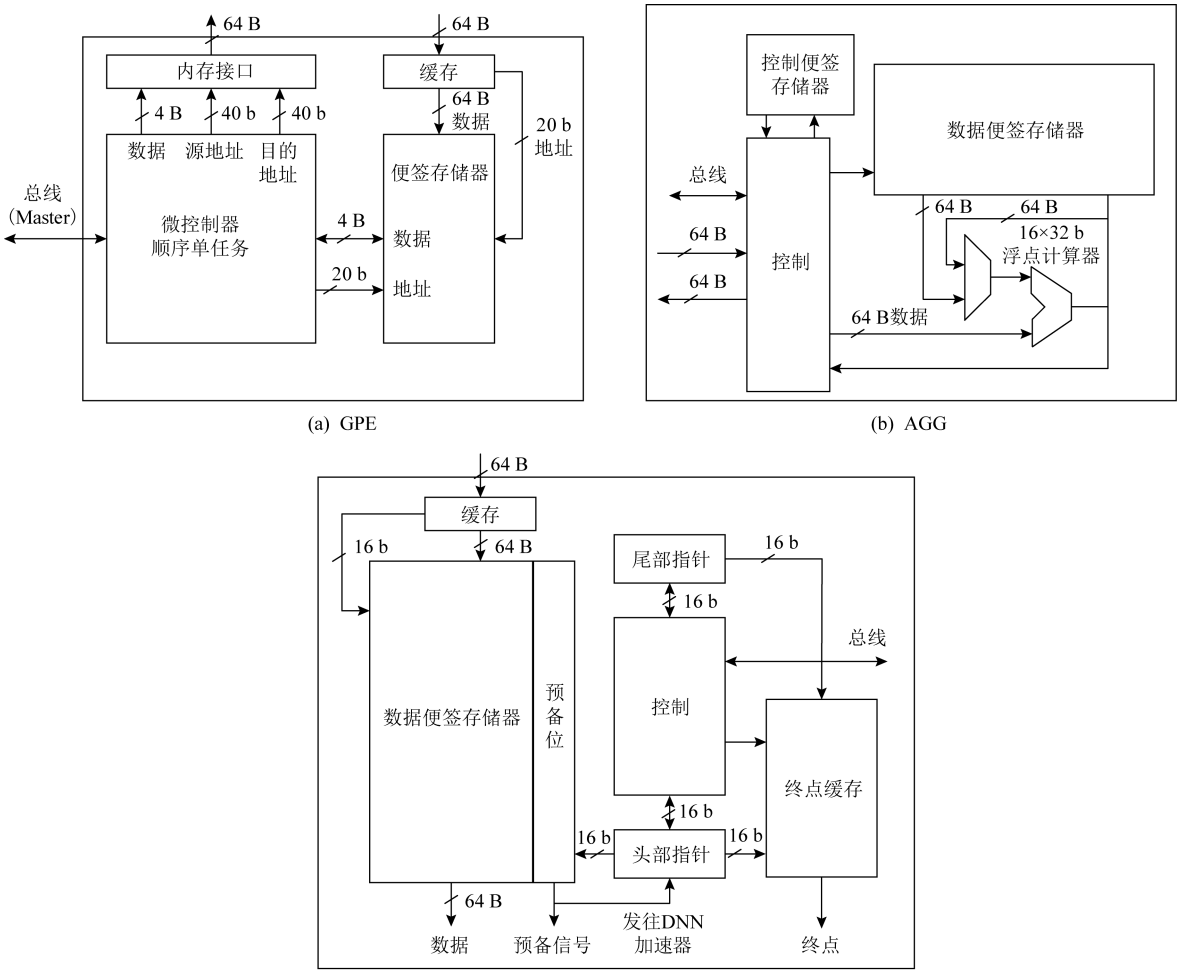


Fig. 9　Module architecture of ref [105]

图9　文献[105]各模块结构图

过程中的状态信息.在 DNN 队列(DNN queue, DNQ)中设置 2 块存储区域,其中大容量片上存储器(62 KB)负责暂存队列数据和延迟准备信息,小容量片上存储器(2 KB)负责存储路由相应的终点信息,大小容量片上存储器相互配合,为 DNN 加速器和不同的 DNN 模型提供运行支持.与 DNQ 相类似的,在聚合器(aggregator, AGG)中添加一对不同容量的片上存储器,其中大容量片上存储器(62 KB)负责暂存聚合过程的中间结果,小容量存储器负责存储每个聚合节点的目的地址信息,用于辅助控制逻辑.

GraphACT[111]利用 FPGA 上大容量的片上存储器 BRAM 来存储图神经网络应用训练过程中的各类数据,具体包括:节点的特征向量、子图的拓扑数据、预处理数据、聚合过程的中间结果数据、梯度信息、模型权重数据、优化器辅助信息等.赛灵思 Alveo 开发板中的一块 BRAM(36 b×1 K),可以存储 1 K 个不同节点的特征数据(32 b 浮点).同时,由于图神经网络的图节点特征数据所需空间巨大,片上存储空间无法完全容纳,因此 GraphACT 采用文献[121]中的方法,通过对图进行采样获取 mini-batch,此方法获取的 mini-batch 包含子图在当前层的完整负载.与完整的负载相比,mini-batch 的数据信息能够更好地适应有限的片上存储空间.

文献[112]首先设计数据分割策略,在预处理的过程中将大规模图分割为若干小的邻接矩阵,通过适当的参数选取,每个数据子块都能够适配存储于 FPGA 的片上 BRAM 中.与上述各图神经网络加速结构设计相类似的,文献[112]加速结构中同样具备多种片上存储器,分别用于存储聚合过程的中间结果、权重矩阵、输出结果等相关数据,另外还在聚合模块中对片上存储采用双缓冲的技术,与数据分割技术相配合,掩盖访存延迟.

EnGN[116]首先采用图分块策略(graph tiling)对图数据进行分块,使得大规模图经过划分,成为适合片上存储规模并最大化局部性的若干子图.同时,配合行导向(row-oriented)或列导向(column-oriented)的数据流处理方向选择,可最大程度重用片上的节点数据,并减少访存开销.但现实世界的图数据规模巨大,导致仅采用图分块策略无法让子图完全适应于处理单元的寄存器堆尺寸以及长延迟的访存,因此 EnGN 还设计了多层次的片上存储器.如图 10 所示,每个 PE 的片上存储器由寄存器堆、度数感知 Cache(degree aware cache, DAVC)和结果 banks 组成,上述三者依次作为 1 级、2 级和 3 级片上存储.DAVC 的空间全部用作缓存高度数节点,且用目的节点的 ID 作为行标签,以确定在 DAVC 是否命中.如果命中,则节点数据会直接从 DAVC 中读出并送往处理单元中的目的节点寄存器中,否则 EnGN 进行 3 级片上访存.

DeepBurning-GL[115]对规则访问的数据和不规则访问的数据进行区分,使用常规的片上存储器来存储规则访问的数据,通过 1 个度数感知 Cache 来存储不规则访问的数据.度数感知 Cache 的核心策略是为高度数节点赋予更高的优先级,不会被频繁地替换.由于相较于低度数节点,高度数节点的数据被重用的可能性更大,因此该策略能够有效提升片上数据的重用率.

Cambricon-G[118]设置混合的片上存储系统,其中包含 1 组便笺存储器(scratchpad memory, SPM)和 1 个拓扑感知 Cache.Cambricon-G 加速结构中共有 3 块 SPM 用于保存规则访存的数据,也即边特征数据、节点的中间或输出特征数据以及权重数据.拓扑感知 Cache 用于保存被不规则访问的数据,也即每层图神经网络输入的节点特征数据,从而提升数据重用率.另外,感知拓扑的意义在于能够应对动态变化的图结构,通过节点重用距离和度数来衡量节点的拓扑,从而实现不同的数据替换策略.

2) 图数据重排序

图数据重排序的核心思想是在数据分割的基础上,通过将邻居节点进行分组和序号重排,使得邻居节点能够分布在同一个分块中,提升图数据的局部性及片上数据的重用率.

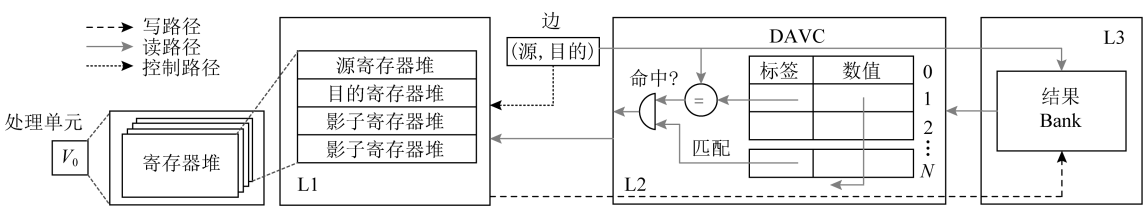


Fig. 10 Three-level on-chip memory of EnGN[112]

图 10 EnGN[112]的 3 级片上存储结构

文献[112]在预处理的过程中,首先对图数据进行稀疏化处理,也即去除冗余边.然后,再对稀疏化的图数据进行重排序处理.具体而言,在原始图中每个节点的序号都是随机的,由于在图聚合阶段,每个节点需要聚合所有邻居节点的信息,因此在重排序的过程中,将节点根据邻接关系进行分组和重排.以图 11 为例,在原始的图分布中,节点随机排布,经过稀疏化处理后的图 G' 中,虽然 2,3,4 号节点构成一个完整子图,但在邻接矩阵 $\boldsymbol{A}$ 中这 3 个节点落在了不同的分块中,导致片上资源不能得到充分利用,并引发更多的片外访存.而经过重排序的预处理过程,G' 中的 2,3,4 号节点在重排形成的图中变为 1,2,3 号节点.如此操作后,这 3 个节点特征的传播和聚合过程都能够在同一个邻接矩阵子块中完成.上述过程通过采用文献[122]中提出的带宽压缩算法(bandwidth reduction algorithm, BR)来实现.数据重排序的方法能够将邻接节点进行分组重排,从而有效提升片上存储资源的重用率.

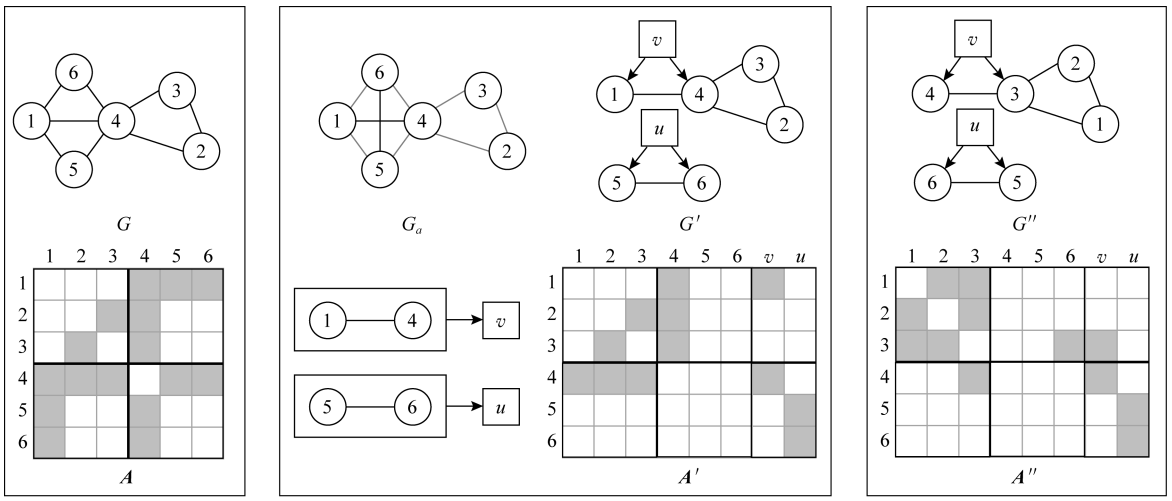


Fig. 11　An example of data preprocessing in ref [112]

图 11　文献[112]的图数据预处理过程举例

3) 降低片上存储需求

FPGAN[114]通过压缩模型权重的方式,降低图注意力网络对片上存储空间的需求,使得有限的片上存储能够承载规模更大的模型.压缩权重的过程分为 2 个步骤:①通过压缩算法将浮点权重转换为 1 组由零或 2 的幂次方组成的定点数;②参照一系列的规则进行数值编码.在此过程中计算精确度缺失,若缺失过大则需要进行重新训练并重复上述步骤.通过数值转换和编码,可使用更小的空间来存储权重数据,从而降低对片上存储空间的需求.

6.3　片外访存优化层次

图神经网络加速结构设计在片外存储层次需要解决粗粒度不规则访存导致的片外访存效率和带宽利用率低下的问题.目前主要的解决思路包括:基于特定的图数据划分方法提高预取效率和数据重用率;利用稀疏性消除、动态访存调度、数据结构重组提高带宽利用率;通过操作融合减少访存带宽需求.

1) 图数据划分

为了提高片外带宽利用率,HyGCN[9]借鉴文献[123-124]中的 Interval 和 Shard 的抽象概念来对图神经网络中的图数据进行划分.每个 Interval 中的节点序号连续,在片外顺序存储,因此这些节点的特征向量可以被连续读进输入缓存中,能够有效提升带宽利用率.

AWB-GCN[104]采用矩阵分块的优化方式(matrix blocking optimization)来提升数据局部性,减少片外的访存带宽需求.如图 12 所示,邻接矩阵 $\boldsymbol{A}$ 被划分为多个子块,对于邻接矩阵 $\boldsymbol{A}$ 与 $\boldsymbol{XW}$(注:$\boldsymbol{X}$ 为特征矩阵,$\boldsymbol{W}$ 为权重矩阵)的矩阵乘运算,不采用矩阵

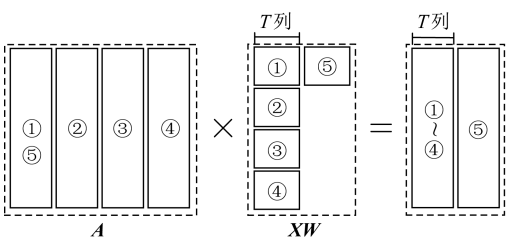


Fig. 12　Matric blocking optimization in AWB-GCN[104]

图 12　AWB-GCN[104]的矩阵分块优化策略

$\boldsymbol{A}$ 的所有子块与 $\boldsymbol{XW}$ 的所有对应列相乘的方式，AWB-GCN 将 $\boldsymbol{A}(\boldsymbol{XW})$ 的 t 列同时并行运算.$\boldsymbol{A}$ 中只有等到前一个子块被重用了 t 次并且完成了 t 列中间结果运算之后，才能开启下一个子块的运算.因此，矩阵 $\boldsymbol{A}$ 的数据重用率得到了 t 倍的提升，从而有效减少了片外访存.

Cambricon-G[118] 提出一种名为邻接立方体(adjacent cuboid)的新型结构来存储图数据.邻接立方体将传统的 2D 邻接矩阵扩展到 3D 空间，每个维度分别代表源节点(*src*)、目的节点(*dst*)和节点的特征(*feat*).邻接立方体根据片上存储空间，被划分为若干小方块(cubelet).Cambricon-G 提出多维度时间分块策略(multi-dimensional temporal tiling)，分别从邻接立方体的 3 个维度进行子块划分，同时实现不同的数据重用：*feat* 维度划分可实现子块间的图拓扑(也即源节点和目的节点之间的边形成的 2D 邻接矩阵)重用；*dst* 维度划分可实现相同源节点所需的节点数据在不同子块间重用；*src* 维度划分可实现节点特征聚合的中间值重用.另外，对于动态变化的图来说，当需要新增/删除边时，仅需对与之相关的子块进行数据搬移.上述策略能够尽可能地挖掘图的局部性，有效提升数据重用率，降低片外访存带宽需求.

2) 稀疏性消除

图神经网络的图数据具有很强的稀疏性，导致了很多无用的片外访存行为.HyGCN[9]，GraphACT[111] 和文献[112]均设计消除图稀疏性的方法来减少片外访存.

HyGCN 为消除稀疏性，提出了一种基于窗口滑动收缩的稀疏性消除机制.窗口尺寸与 Shard 一致.对于每个节点 Interval，窗口逐渐向下滑动，直到有边出现在窗口最顶行才会停止.然后，从该窗口底部行的相邻行开始创建下一个新窗口，每个窗口的停止条件都相同.执行过程中不断生成新窗口，并向下滑动.为减少固定大小的窗口底部存在的稀疏，HyGCN 在每个窗口滑动结束之后还会进行窗口收缩.具体而言，每个窗口从底部行向上收缩，直到底部行遇到有效边为止.由于窗口收缩，最终 Shard 的大小通常会有所不同.每个窗口中访存获取的邻居节点特征，能够被多个节点的聚合操作的输入所共享，也进一步降低了片外访存需求.上述窗口滑动收缩的过程，动态消除图神经网络中图数据的稀疏性，从而减少冗余的片外访存操作.

GraphACT[111] 和文献[112]同样设计图数据稀疏性消除机制作为数据预处理的其中一个步骤.其主要思想是都是通过消除图中的冗余边来消除稀疏性，同时保证不影响最终的处理结果.文献[112]采用 GraphACT 中提出的冗余缩减方法来消除边的冗余，与 6.1 节中所述的减少冗余计算方法一致.通过减少冗余边，不仅能够优化计算，同时也能有效减少冗余的片外访存需求.图 11 为文献[112]的稀疏性消除与处理过程的简单示例.对于图 G，首先枚举每个节点的邻居对，例如 1 号节点有邻居 4,5,6 号节点，也即枚举 1 号节点的邻居对为(4,5)，(4,6)和(5,6).其中(5,6)邻居对是被 1 号和 4 号节点共享的邻居对.类似地，G 图中共有(1,4)和(5,6)两个共用邻居对.接着将共用邻居对替换为新的节点 u 和 v.将 u 和 v 分别与特征向量 $\frac{1}{2}(\boldsymbol{X}(5)+\boldsymbol{X}(6))$ 和 $\frac{1}{2}(\boldsymbol{X}(1)+\boldsymbol{X}(4))$ 相连接，然后合并新节点，删除冗余的边，形成新的图 G'.还可通过多个迭代反复执行这一过程来进一步减少冗余.

3) 动态访存调度

在实际的应用场合中，HyGCN[9] 的 Aggregation 加速引擎和 Combination 加速引擎需要的片外带宽因图神经网络模型的差异而不同，因此很难在设计阶段确定 2 个加速引擎之间的内存带宽比率.HyGCN 使用统一的片外存储为 2 个加速引擎供应数据，并同时提供基于优先级的动态内存调度策略.Aggregation 加速引擎的片外访存包括输入特征向量和边数据，Combination 加速引擎的片外访存包括权重读入和输出向量写回.这些地址不连续的访存请求同时进行，会导致 DRAM 行缓存命中率低下.因此 HyGCN 预定义了访问优先级，根据访存优先级重新组织这些不连续的请求为不同的批次，进而可以逐批执行访存请求.另外，当前批次中低优先级的请求也会先于下一批次的高优先级请求执行，也即当前批处理中的低优先级访问将在下一批高优先级访问之前进行处理，而不是始终首先进行高优先级访问.上述策略可以显著提高 DRAM 的行缓存利用率，从而改善片外访存行为.此外，被连续请求的数据会被分到不同的 bank 上，以挖掘 DRAM 的 bank 级并行性.

4) 数据结构重组

针对现实图数据的稀疏性问题，FPGAN[114] 提出一种数据结构重组的策略，在表示图结构的传统邻接列表基础上，对图数据进行向量化和对齐处理，从而实现更加高效的数据片外访存.图 13 展示了节点

数据实施该策略时的一个示例,其中矩阵的行代表节点的特征向量,NV 是节点向量的尺寸,FV 代表特征向量的尺寸.为了能让行数被 NV 整除以及列数被 FV 整除,该工作对矩阵向右和向下进行填充零的操作.图 13 中的方向代表数据是从左向右以及从上到下进行向量化的.该策略完成后的输出是一个一维数组,每个向量块的大小为 next_power_of_2($NV\times FV$),其中 next_power_of_2(v)计算的数值大于等于 v,并且是 2 的最小次方,以此对齐数据.除了节点特征向量之外,该策略还支持对权重数据、邻接列表的编号等数据进行操作.

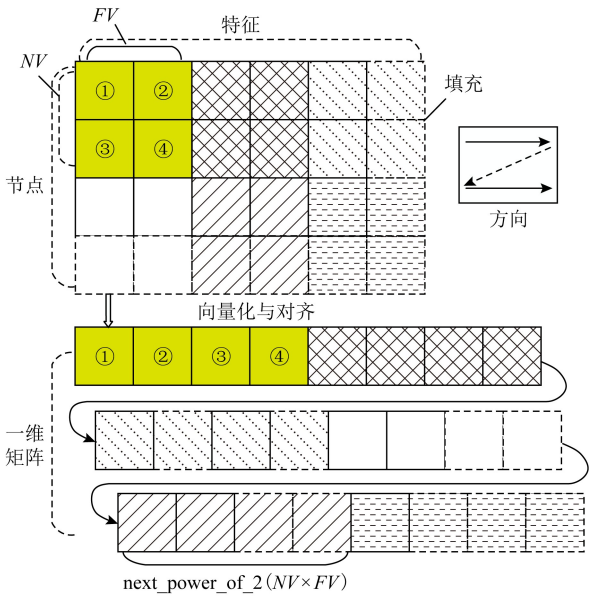


Fig. 13　Illustration of data structure reorganization in FPGAN[114]

图 13　FPGAN[114]的数据结构重组策略示意图

5) 操作融合

对于复杂的自注意力机制(self-attention mechanism),FPGAN[114]通过操作融合(operation fusion)的方式剔除其中的存储同步过程,从而节省片外访存带宽.具体而言,FPGAN 将图注意力网络的每一层拆分为特征聚合和线性转换 2 个阶段,将自注意力机制拆分为计算和归一化注意力系数 2 个步骤.FPGAN 首先通过将自注意力机制中的共享权重 a 拆分为用于中心节点的权重 a_1 和用于相邻节点的权重 a_2 来修改注意力系数的计算方式.然后将注意力系数的计算过程映射入线性转换阶段,将注意力系数的归一化映射入特征聚合阶段.上述过程完成自注意力机制与通用图注意力网络两阶段的操作融合,从而剔除自注意力机制中的存储同步过程,降低片外访存带宽需求.

6) 存内计算

GNN-PIM[117]是首款为图神经网络应用定制的存内计算加速结构,其利用阻变式随机存储器(resistive random access memory, ReRAM)实现存内计算.如图 14 所示,GNN-PIM 加速结构由多个节点簇(node cluster)构成,每个节点簇由若干用于处理图节点或边的节点组成,而每个节点中同时包含计算单元和存储单元,PIM 的设计缩短了计算单元与存储单元之间的距离,并能提供更多的空间用于存储图数据.同时为高效进行节点间的数据交互,GNN-PIM 设计分层片上网络(network on chip, NoC)实现节点簇之间及节点簇内部节点之间的通信,为数据传输提供更高的带宽.另外 PIM 的设计还能为访存带宽带来更多可扩展性空间.

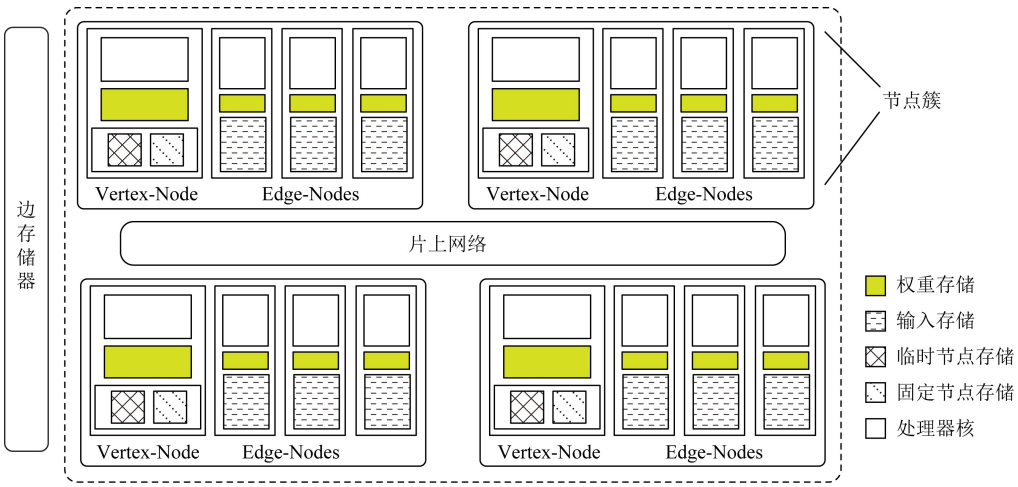


Fig. 14　Overall architecture of GNN-PIM[117]

图 14　GNN-PIM[117]的整体结构图

7 总结与展望

图神经网络是时下学术界和工业界最热门的研究方向之一,本文首先对图神经网络应用的基础知识、常见算法、应用场景、编程模型以及主流的基于通用平台的框架与扩展库等进行介绍.然后以图神经网络执行行为带来的加速结构设计挑战为出发点,从整体结构设计以及计算、片上访存、片外访存多个层次对该领域的关键优化技术进行详实而系统的分析与介绍.

由于图神经网络加速结构研究尚处于新兴初始研究阶段,其仍具备很大的发展优化空间,具体而言有 4 个方面:

1) 大规模多节点加速结构.随着大数据时代图数据规模的不断上升,图神经网络应用对计算资源的需求也愈加提高,单节点将难以高效执行超大规模图神经网络应用,算力甚至可能无法满足其运行需求.因此设计可扩展的多节点加速系统势在必行,目前学术界和工业界尚未有该类成果问世,相信在不久的未来,该方向会成为图神经网络应用的研究热点之一.

2) 异质图神经网络加速结构.现实生活中相对同质图,异质图是更为常见的图结构,异质图中节点和边可以有多种不同的类别,且节点相连关系更为复杂,但数据表达能力更强,适用场景更广.尽管已有针对异质图的图神经网络算法实现,DGL 框架也开始支持异质图的构建,但面向异质图神经网络应用的加速结构领域仍是无人区.

3) 算法与阶段支持灵活化.目前图神经网络专用加速结构多只针对单类别算法进行加速支持.但随着图神经网络算法的高速发展,不仅算法种类会有所增加,并且每种类别中的具体算法也会不断革新,这就使得加速结构是否能高效适应与支持后续的新型算法成为一个亟需克服的难题.另外,目前的已有工作大多只能针对训练或者推断中的单一阶段进行加速,如何能够让加速结构高效地同时对 2 个阶段产生加速作用也是一个值得思考的问题.因此提升图神经网络加速结构设计对算法与执行阶段支持的灵活性是另一个极具应用价值的研究方向.

4) 图神经网络加速结构产业化落地.图神经网络作为推动人工智能领域革新、迈入“认知智能”时代的核心主力军,极富商业应用价值,有望为人们生活智能化与科技进步做出更大的贡献.有寒武纪等在内的国内自主研发的人工智能芯片珠玉在前,相信面向图神经网络应用的产业化芯片也指日可待,且必将创造更多辉煌.

图神经网络作为推动认知智能时代发展的重要应用之一,具有极高的研究价值与产业前景.相信本文对面向图神经网络应用的加速结构设计中涉及到的关键优化技术与未来方向的详实介绍,能够让读者清晰地了解该领域的研究现状,并对相关研究人员在加速结构设计过程中有所启发.

参 考 文 献

[1] Wu Zonghan, Pan Shirui, Chen Fengwen, et al. A comprehensive survey on graph neural networks [J]. IEEE Transactions on Neural Networks and Learning Systems, 2012, 32(1): 4-24

[2] Gui Chuangyi, Zheng Long, He Bingsheng, et al. A survey on graph processing accelerators: Challenges and opportunities [J]. Journal of Computer Science and Technology, 2019, 34(2): 339-371

[3] Kipf T N, Welling M. Semi-supervised classification with graph convolutionalnetworks [J]. arXiv preprint arXiv:1609.02907, 2016

[4] Dai Hanjun, Kozareva Z, Dai Bo, et al. Learning steady-states of iterative algorithms over graphs [C] //Proc of Int Conf on Machine Learning. New York: Association for Computing Machinery, 2018: 1106-1114

[5] Liu Ziqi, Chen Chaochao, Yang Xinxing, et al. Heterogeneous graph neural networks for malicious account detection [C] //Proc of the 27th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2018: 2077-2085

[6] Yan Mingyu, Ye Xiaochun, Fan Dongrui. Graph neural network acceleration chip: Propellant for the takeoff of artificial intelligence “cognitive Intelligence” [J]. Communications of the CCF, 2020, 16(10): 36-44 (in Chinese)
(严明玉, 叶笑春, 范东睿. 图神经网络加速芯片:人工智能“认知智能”阶段起飞的推动剂[J]. 中国计算机学会通讯, 2020, 16(10): 36-44)

[7] Alibaba DAMO Academy. Top 10 Tech Trends of 2019 [EB/OL].[2021-02-20]. https://damo.alibaba.com/events/50 (in Chinese)
(阿里巴巴达摩院. 达摩院 2019 十大科技趋势[EB/OL]. [2021-02-20]. https://damo.alibaba.com/events/50)

[8] Alibaba DAMO Academy. Top 10 Tech Trends of 2020 [EB/OL]. [2021-02-20]. https://damo.alibaba.com/events/57 (in Chinese)
(阿里巴巴达摩院. 达摩院 2020 十大科技趋势[EB/OL]. [2021-02-20]. https://damo.alibaba.com/events/57)

[9] Yan Mingyu, Deng Lei, Hu Xing, et al. HyGCN: A GCN accelerator with hybrid architecture [C] //Proc of 2020 IEEE Int Symp on High Performance Computer Architecture (HPCA). Piscataway, NJ: IEEE, 2020: 15-29

[10] Zhang Ziwei, Cui Peng, Zhu Wenwu. Deep learning on graphs: A survey [J/OL]. IEEE Transactions on Knowledge and Data Engineering, 2020[2021-02-20]. https://ieeexplore.ieee.org/document/9039675/citations?tabFilter=papers

[11] Hamilton W L, Ying R, Leskovec J. Representation learning on graphs: Methods andapplications [J]. arXiv preprint arXiv:1709.05584, 2017

[12] Sato R. A survey on the expressive power of graph neuralnetworks [J]. arXiv preprint arXiv:2003.04078, 2020

[13] Yan Mingyu, Li Han, Deng Lei, et al. A survey on graph processing accelerators [J]. Journal of Computer Research and Development, 2021, 58(4): 862-887 (in Chinese)
(严明玉, 李涵, 邓磊, 等. 图计算加速架构综述[J]. 计算机研究与发展, 2021, 58(4): 862-887)

[14] Wang Jizhe, Huang Pipei, Zhao Huan, et al. Billion-scale commodity embedding for e-commerce recommendation in Alibaba [C] //Proc of the 24th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2018: 839-848

[15] Ching A, Edunov S, Kabiljo M, et al. One trillion edges: Graph processing at facebook-scale [J]. Proceedings of the VLDB Endowment, 2015, 8(12): 1804-1815

[16] Ying Rex, He Ruining, Chen Kaifeng, et al. Graph convolutional neural networks for web-scale recommender systems [C] //Proc of the 24th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2018: 974-983

[17] Hamilton W L. Graph representationlearning [J]. Synthesis Lectures on Artifical Intelligence and Machine Learning, 2020, 14(3): 1-159

[18] Werbos P J. Backpropagation through time: What it does and how to doit [J]. Proceedings of the IEEE, 1990, 78(10): 1550-1560

[19] Pineda F J. Generalization of back-propagation to recurrent neuralnetworks [J]. Physical Review Letters, 1987, 59(19): 2229

[20] Almeida L B. ALearning Rule for Asynchronous Perceptrons with Feedback in a Combinatorial Environment [M]. Los Alamitos, CA: IEEE Computer Society, 1990: 102-111

[21] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors [J]. Nature, 1986, 323(6088): 533-536

[22] Zhang Muhan, Cui Zhicheng, Neumann M, et al. An end-to-end deep learning architecture for graph classification [C] // Proc of the AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2018: 4438-4445

[23] Ying R, You Jiaxuan, Morris C, et al. Hierarchical graph representation learning with differentiable pooling [C] // Advances in Neural Information Processing Systems: Annual Conf on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2018: 4801-4811

[24] Pan Shirui, Wu Jia, Zhu Xingquan, et al. Joint structure feature exploration and regularization for multi-task graph classification [J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 28(3): 715-728

[25] Pan Shirui, Wu Jia, Zhu Xingquan, et al. Task sensitive feature exploration and learning for multitask graph classification [J]. IEEE Transactions on Cybernetics, 2016, 47(3): 744-758

[26] Kipf T N, Welling M. Variational graph auto-encoders [J]. arXiv preprint arXiv:1611.07308, 2016

[27] Pan Shirui, Hu Ruiqi, Long Guodong, et al. Adversarially regularized graph autoencoder for graph embedding [J]. arXiv preprint arXiv:1802.04407, 2018

[28] Hamilton W, Ying R, Leskovec J. Inductive representation learning on large graphs [C] //Advances in Neural Information Processing Systems. San Francisco, CA: Morgan Kaufmann, 2017: 1024-1034

[29] Chen Jie, Ma Tengfei, Xiao Cao. FastGCN: Fast learning with graph convolutional networks via importance sampling [J]. arXiv preprint arXiv:1801.10247, 2018

[30] Huang Wenbing, Zhang Tong, Rong Yu, et al. Adaptive sampling towards fast graph representation learning [J]. arXiv preprint arXiv:1809.05343, 2018

[31] Veličković P, Fedus W, Hamilton W L, et al. Deep graph infomax [J]. arXiv preprint arXiv:1809.10341, 2018

[32] Hu Ziniu, Dong Yuxiao, Wang Kuansan, et al. Gpt-GNN: Generative pre-training of graph neural networks [C] //Proc of the 26th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2020: 1857-1867

[33] He Kaiming, Zhang Xianyu, Ren Shaoqing, et al. Deep residual learning for image recognition [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 770-778

[34] Zhou Jie, Cui Ganqu, Hu Shengding, et al. Graph neural networks: A review of methods andapplications [J]. arXiv preprint arXiv:1812.08434, 2018

[35] Li Ruoyu, Wang Sheng, Zhu Feiyun, et al. Adaptive graph convolutional neural networks [C] //Proc of the AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2018, 32(1): 3546-3553

[36] Xu Keyulu, Hu Weihua, Leskovec J, et al. How powerful are graph neural networks? [J]. arXiv preprint arXiv:1810.00826, 2018

[37] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks [J]. arXiv preprint arXiv:1710.10903, 2017

[38] Zhang Jiani, Shi Xingjian, Xie Junyuan, et al. Gaan: Gated attention networks for learning on large and spatiotemporal graphs [J]. arXiv preprint arXiv:1803.07294, 2018

[39] Li Yujia, Tarlow D, Brockschmidt M, et al. Gated graph sequence neural networks [J]. arXiv preprint arXiv:1511.05493, 2015

[40] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation [J]. arXiv preprint arXiv: 1406.1078, 2014

[41] Vincent P, Larochelle H, Bengio Y, et al. Extracting and composing robust features with denoising autoencoders [C] // Proc of the 25th Int Conf on Machine Learning. New York: Association for Computing Machinery, 2008: 1096-1103

[42] Tu Ke, Cui Peng, Wang Xiao, et al. Deep recursive network embedding with regular equivalence [C] //Proc of the 24th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2018: 2357-2366

[43] Jain A, Zamir A R, Savarese S, et al. Structural-RNN: Deep learning on spatio-temporal graphs [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 5308-5317

[44] Wu Zonghan, Pan Shirui, Long Guodong, et al. Graph wavenet for deep spatial-temporal graph modeling [J]. arXiv preprint arXiv:1906.00121, 2019

[45] Wu Shiwen, Zhang Wentao, Sun Fei, et al. Graph neural networks in recommender systems: A survey [J]. arXiv preprint arXiv:2011.02260, 2020

[46] Berg R, Kipf T N, Welling M. Graph convolutional matrix completion [J]. arXiv preprint arXiv:1706.02263, 2017

[47] Zheng Lei, Lu Chunta, Jiang Fei, et al. Spectral collaborative filtering [C] //Proc of the 12th ACM Conf on Recommender Systems. New York: ACM, 2018: 311-319

[48] Wang Xiang, He Xiangnan, Wang Meng, et al. Neural graph collaborative filtering [C] //Proc of the 42nd Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2019: 165-174

[49] Wu Le, Sun Peijie, Fu Yanjie, et al. A neural influence diffusion model for social recommendation [C] //Proc of the 42nd Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2019: 235-244

[50] Wu Qitian, Zhang Hengrui, Gao Xiaofeng, et al. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems [C] // Proc of the World Wide Web Conf. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2019: 2091-2102

[51] Fan Wenqi, Ma Yao, Li Qing, et al. Graph neural networks for social recommendation [C] //Proc of the World Wide Web Conf. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2019: 417-426

[52] Wang Hongwei, Zhao Miao, Xie Xing, et al. Knowledge graph convolutional networks for recommender systems [C] //Proc of the World Wide Web Conf. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2019: 3307-3313

[53] Wang Xiang, He Xiangnan, Cao Yixin, et al. KGAT: Knowledge graph attention network for recommendation [C] //Proc of the 25th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2019: 950-958

[54] Song Weiping, Xiao Zhiping, Wang Yifan, et al. Session-based social recommendation via dynamic graph attention networks [C] //Proc of the 12th ACM Int Conf on Web Search and Data Mining. New York: ACM, 2019: 555-563

[55] Wu Shu, Tang Yuyuan, Zhu Yanqiao, et al. Session-based recommendation with graph neural networks [C] //Proc of the AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2019, 33(1): 346-353

[56] Kampffmeyer M, Chen Yinbo, Liang Xiaodan, et al. Rethinking knowledge graph propagation for zero-shot learning [C] // Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 11487-11496

[57] Wang Xiaolong, Ye Yufei, Gupta A. Zero-shot recognition via semantic embeddings and knowledge graphs [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 6857-6866

[58] Garcia V, Bruna J. Few-shot learning with graph neural networks [J]. arXiv preprint arXiv:1711.04043, 2017

[59] Marino K, Salakhutdinov R, Gupta A. The more you know: Using knowledge graphs for imageclassification [J]. arXiv preprint arXiv:1612.04844, 2016

[60] Lee C W, Fang Wei, Yeh C K, et al. Multi-label zero-shot learning with structured knowledge graphs [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 1576-1585

[61] Teney D, Liu Lingqiao, van Den Hengel A. Graph-structured representations for visual question answering [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 1-9

[62] Norcliffe-Brown W, Vafeias E, Parisot S. Learning conditioned graph structures for interpretable visual question answering [J]. arXiv preprint arXiv:1806.07243, 2018

[63] Narasimhan M, Lazebnik S, Schwing A G. Out of the box: Reasoning with graph convolution nets for factual visual question answering [J]. arXiv preprint arXiv: 1811.00538, 2018

[64] Wang Zhouxia, Chen Tianshui, Ren Jimmy, et al. Deep reasoning with knowledge graph for social relationship understanding [J]. arXiv preprint arXiv:1807.00504, 2018

[65] Si Chenyang, Jing Ya, Wang Wei, et al. Skeleton-based action recognition with spatial reasoning and temporal stack learning [C] //Proc of the European Conf on Computer Vision (ECCV). Berlin: Springer, 2018: 103-118

[66] Peng Wei, Hong Xiaopeng, Chen Haoyu, et al. Learning graph convolutional network for skeleton-based human action recognition by neural searching [C] //Proc of the AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2020, 34(3): 2669-2676

[67] Su Kun, Liu Xiulong, Shlizerman E. Predict & cluster: Unsupervised skeleton based action recognition [C] //Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 9631-9640

[68] Hao Xiaoke, Li Jie, Guo Yingchun, et al. Hypergraph neural network for skeleton-based action recognition [J]. IEEE Transactions on Image Processing, 2021, 30: 2263-2275

[69] Landrieu L, Simonovsky M. Large-scale point cloud semantic segmentation with superpoint graphs [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 4558-4567

[70] Te Gusi, Hu Wei, Guo Zongming, et al. RGCNN: Regularized graph CNN for point cloud segmentation [C] // Proc of the 26th ACM Int Conf on Multimedia. New York: ACM, 2018: 746-754

[71] Wang Yue, Sun Yongbin, Liu Ziwei, et al. Dynamic graph CNN for learning on point clouds [J]. ACM Transactions on Graphics, 2019, 38(5): 1-12

[72] Peng Hao, Li Jianxin, He Yu, et al. Large-scale hierarchical text classification with recursively regularized deep graph-CNN [C] //Proc of the 2018 World Wide Web Conf. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2018: 1063-1072

[73] Zhang Yue, Liu Qi, Song Linfeng. Sentence-state LSTM for text representation [J]. arXiv preprint arXiv: 1805.02474, 2018

[74] Yao Liang, Mao Chengsheng, Luo Yuan. Graph convolutional networks for text classification [C] //Proc of the AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2019, 33(1): 7370-7377

[75] Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks [J]. arXiv preprint arXiv:1503.00075, 2015

[76] Zhang Ningyu, Deng Shumin, Sun Zhanlin, et al. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks [J]. arXiv preprint arXiv:1903.01306, 2019

[77] Zhang Yan, Guo Zhijiang, Lu Wei. Attention guided graph convolutional networks for relation extraction [J]. arXiv preprint arXiv:1906.07510, 2019

[78] Zhu Hao, Lin Yankai, Liu Zhiyuan, et al. Graph neural networks with generated parameters for relation extraction [J]. arXiv preprint arXiv:1902.00756, 2019

[79] Fu T J, Li P H, Ma Weiyun. GraphRel: Modeling text as relational graphs for joint entity and relation extraction [C] // Proc of the 57th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: Association for Computational Linguistics, 2019: 1409-1418

[80] Chen Bo, Sun Le, Han Xianpei. Sequence-to-action: End-to-end semantic graph generation for semantic parsing [J]. arXiv preprint arXiv:1809.00773, 2018

[81] Battaglia P W, Pascanu R, Lai M, et al. Interaction networks for learning about objects, relations and physics [J]. arXiv preprint arXiv:1612.00222, 2016

[82] Sukhbaatar S, Szlam A, Fergus R. Learning multiagent communication withbackpropagation [J]. arXiv preprint arXiv:1605.07736, 2016

[83] Hoshen Y. Vain: Attentional multi-agent predictivemodeling [J]. arXiv preprint arXiv:1706.06122, 2017

[84] Duvenaud D, Maclaurin D, Aguilera-Iparraguirre J, et al. Convolutional networks on graphs for learning molecular fingerprints [J]. arXiv preprint arXiv:1509.09292, 2015

[85] Kearnes S, McCloskey K, Berndl M, et al. Molecular graph convolutions: Moving beyond fingerprints [J]. Journal of Computer-Aided Molecular Design, 2016, 30(8): 595-608

[86] Jin W, Barzilay R, Jaakkola T. Junction tree variational autoencoder for molecular graph generation [C] //Proc of Int Conf on Machine Learning. New York: Association for Computing Machinery, 2018: 2323-2332

[87] Fout A M. Protein interface prediction using graph convolutional networks [D]. Colorado: Colorado State University, 2017

[88] Xu Nuo, Wang Pinghui, Chen Long, et al. Mr-GNN: Multi-resolution and dual graph neural network for predicting structured entity interactions [J]. arXiv preprint arXiv:1905.09558, 2019

[89] Bazzan A L C, Klügl F. Introduction to intelligent systems in traffic and transportation [J]. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2013, 7(3): 1-137

[90] Li Yaguang, Yu Rose, Shahabi C, et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting [J]. arXiv preprint arXiv:1707.01926, 2017

[91] Yu Bing, Yin Haoteng, Zhu Zhanxing. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting [J]. arXiv preprint arXiv:1709.04875, 2017

[92] Mengzhang L, Zhanxing Z. Spatial-temporal fusion graph neural networks for traffic flow forecasting [J]. arXiv preprint arXiv:2012.09641, 2020

[93] Chen K, Chen F, Lai Baisheng, et al. Dynamic spatio-temporal graph-based CNNs for traffic flow prediction [J]. IEEE Access, 2020, 8: 185136-185145

[94] Yao Huaxiu, Wu Fei, Ke Jintao, et al. Deep multi-view spatial-temporal network for taxi demand prediction [C] // Proc of the AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2018, 32(1): 2588-2595

[95] Cachay S R, Erickson E, Bucker A F C, et al. Graph neural networks for improved El Ni\～ no forecasting [J]. arXiv preprint arXiv:2012.01598, 2020

[96] Yang Shuo, Zhang Zhiqiang, Zhou Jun, et al. Financial risk analysis for SMEs with graph-based supply chain mining [C] //Proc of the 29th Int Joint Conf on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2020: 4661-4667

[97] Li Xiaoxiao, Saude J, Reddy P, et al. Classifying and understanding financial data using graph neural network [C/OL] //Proc of the AAAI Workshop on Knowledge Discovery from Unstructured Data in Financial Services. Palo Alto, CA: AAAI Press, 2020 [2020-02-20]. https://aaai-kdf2020.github.io/assets/pdfs/kdf2020_paper_21.pdf

[98] Hu linmei, Yang Tianchi, Shi Chuan. Research progress of knowledge graph based on graph neural network [J]. Communications of the CCF, 2020, 16(8): 38-48 (in Chinese)
(胡琳梅,杨天持,石川. 基于图神经网络的知识图谱研究进展[J]. 中国计算机学会通讯, 2020, 16(8): 38-48)

[99] Ying R, Bourgeois D, You Jiaxuan, et al. GNNExplainer: Generating explanations for graph neural networks [J]. Advances in Neural Information Processing Systems, 2019, 32: 9240-9251

[100] Yuan Hao, Tang Jiliang, Hu Xia, et al. XGNN: Towards model-level explanations of graph neural networks [C] //Proc of the 26th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2020: 430-438

[101] Zhu Rong, Zhao Kun, Yang Hongxia, et al. AliGraph: A comprehensive graph neural network platform [C] //Proc of the 25th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2019: 3165-3166

[102] Fey M, Lenssen J E.Fast graph representation learning with PyTorch Geometric [J]. arXiv preprint arXiv:1903.02428, 2019

[103] Wang Minjie, Yu Lingfan, Zheng Da, et al. Deep graph library: Towards efficient and scalable deep learning on graphs [J]. arXiv preprint arXiv:1909.01315, 2019

[104] Geng Tong, Li A, Shi Runbin, et al. AWB-GCN: A graph convolutional network accelerator with runtime workload rebalancing [C] //Proc of 2020 53rd Annual IEEE/ACM Int Symp on Microarchitecture (MICRO). Piscataway, NJ: IEEE, 2020: 922-936

[105] Auten A, Tomei M, Kumar R. Hardware acceleration of graph neural networks [C] //Proc of 2020 57th ACM/IEEE Design Automation Conf (DAC). Piscataway, NJ: IEEE, 2020: 1-6

[106] Zhang Zhihui, Leng Jingwen, Ma Lingxiao, et al. Architectural implications of graph neural networks [J]. IEEE Computer Architecture Letters, 2020, 19(1): 59-62

[107] Kersting K, Kriege N M, Morris C, et al. Benchmark data sets for graph kernels [EB/OL]. 2016 [2021-02-20]. http://graphkernels.cs.tu-dortmund.de

[108] Dahlgren F, Dubois M, Stenstrom P. Sequential hardware prefetching in shared-memorymultiprocessors [J]. IEEE Transactions on Parallel and Distributed Systems, 1995, 6(7): 733-746

[109] Yan Mingyu, Chen Zhaodong, Deng Lei, et al. Characterizing and understanding GCNs on GPU [J]. IEEE Computer Architecture Letters, 2020, 19(1): 22-25

[110] Yan Mingyu, Hu Xing, Li Shuangchen, et al. Balancing memory accesses for energy-efficient graph analytics accelerators [C] //Proc of 2019 IEEE/ACM Int Symp on Low Power Electronics and Design (ISLPED). Piscataway, NJ: IEEE, 2019: 1-6

[111] Zeng Hanqing, Prasanna V. GraphACT: Accelerating GCN training on CPU-FPGA heterogeneous platforms [C] //Proc of the 2020 ACM/SIGDA Int Symp on Field-Programmable Gate Arrays. New York: ACM, 2020: 255-265

[112] Zhang Bingyi, Zeng Hanqing, Prasanna V. Hardware acceleration of large scale GCN inference [C] //Proc of 2020 IEEE 31st Int Conf on Application-Specific Systems, Architectures and Processors (ASAP). Piscataway, NJ: IEEE, 2020: 61-68

[113] Cheng Qixuan, Wen Mei, Shen Junzhong, et al. Towards a deep-pipelined architecture for accelerating deep GCN on a multi-FPGA platform [C] //Proc of Int Conf on Algorithms and Architectures for Parallel Processing. Berlin: Springer, 2020: 528-547

[114] Yan Wei'an, Tong Weiqin, Zhi Xiaoli. FPGAN: An FPGA accelerator for graph attention networks with software and hardware co-optimization [J]. IEEE Access, 2020, 8: 171608-171620

[115] Liang Shengwen, Liu Cheng, Wang Ying, et al. DeepBurning-GL: An automated framework for generating graph neural network accelerators [C] //Proc of 2020 IEEE/ACM Int Conf on Computer Aided Design (ICCAD). Piscataway, NJ: IEEE, 2020: 1-9

[116] Liang Shengwen, Wang Ying, Liu Cheng, et al. EnGN: A high-throughput and energy-efficient accelerator for large graph neural networks [J/OL]. IEEE Transactions on Computers, 2020 [2021-02-20]. https://ieeexplore.ieee.org/document/9161360

[117] Wang Zhao, Guan Yijin, Sun Guangyu, et al. GNN-PIM: A processing-in-memory architecture for graph neural networks [C] //Proc of Conf on Advanced Computer Architecture. Berlin: Springer, 2020: 73-86

[118] Song Xinkai, Zhi Tian, Fan Zhe, et al. Cambricon-G: A Polyvalent energy-efficient accelerator for dynamic graph neural networks [J/OL]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2021 [2021-02-20]. https://ieeexplore.ieee.org/document/9326339

[119] Chen Fengwen, Pan Shirui, Jiang Jing, et al. DAGCN: Dual attention graph convolutional networks [C] //Proc of 2019 Int Joint Conf on Neural Networks (IJCNN). Piscataway, NJ: IEEE, 2019: 1-8

[120] Jouppi N P, Young C, Patil N, et al. In-datacenter performance analysis of a tensor processing unit [C] //Proc of the 44th Annual Int Symp on Computer Architecture. New York: ACM, 2017: 1-12

[121] Zeng Hanqing, Zhou Hongkuan, Srivastava A, et al. Accurate, efficient and scalable graph embedding [C] //Proc of 2019 IEEE Int Parallel and Distributed Processing Symp (IPDPS). Piscataway, NJ: IEEE, 2019: 462-471

[122] Chen Jianfei, Zhu Jun, Song Le. Stochastic training of graph convolutional networks with variance reduction [J]. arXiv preprint arXiv:1710.10568, 2017

[123] Kyrola A, Blelloch G, Guestrin C. GraphChi: Large-scale graph computation on just a PC [C] //Proc of the 10th USENIX Symp on Operating Systems Design and Implementation (OSDI 12). Berkeley, CA: USENIX Association, 2012: 31-46

[124] Chi Yuze, Dai Guohao, Wang Yu, et al. Nxgraph: An efficient graph processing system on a single machine [C] // Proc of 2016 IEEE 32nd Int Conf on Data Engineering (ICDE). Piscataway, NJ: IEEE, 2016: 409-420

Li Han, born in 1994. PhD candidate. Student member of CCF. Her main research interests include computer architecture and graph-based hardware accelerator.

李　涵,1994 年生. 博士研究生,CCF 学生会员. 主要研究方向为计算机体系结构和基于图的硬件加速器研究.

Yan Mingyu, born in 1990. PhD candidate. Student member of CCF. His main research interests include graph-based hardware accelerator and dataflow architecture.

严明玉,1990 年生.博士研究生,CCF 学生会员.主要研究方向为基于图的硬件加速器和数据流架构研究.

Lü Zhengyang, born in 1997. PhD candidate. Student member of CCF. Her main research interests include computer architecture and graph-based hardware accelerator. (lvzhengyang19@mails.ucas.ac.cn)

吕征阳,1997 年生.博士研究生,CCF 学生会员.主要研究方向为计算机体系结构和基于图的硬件加速器研究.

Li Wenming, born in 1988. PhD, associate professor. His main research interests include high throughput computing architecture and software simulation. (liwenming@ict.ac.cn)

李文明,1988 年生.博士,副研究员.主要研究方向为高吞吐量计算体系结构和软件仿真.

Ye Xiaochun, born in 1981. PhD, professor. Member of CCF. His main research interests include software simulation, algorithm paralleling and optimizing, and architecture for high-performance computer. (yexiaochun@ict.ac.cn)

叶笑春,1981 年生.博士,研究员,CCF 会员.主要研究方向为软件仿真、算法并行和优化、高性能计算机架构研究.

Fan Dongrui, born in 1979. PhD, professor, PhD supervisor. Distinguished member of CCF. His main research interests include many-core processor design, high throughput processor design and low power micro-architecture. (fandr@ict.ac.cn)

范东睿,1979 年生.博士,研究员,博士生导师,CCF 杰出会员.主要研究方向为众核处理器设计、高通量处理器设计和低功耗微架构研究.

Tang Zhimin, born in 1966. PhD, professor. Distinguished member of CCF. His main research interests include high performance computer architecture, processor design and digital signal processing. (tang@ict.ac.cn)

唐志敏,1966 年生.博士,研究员,CCF 杰出会员.主要研究方向为高性能计算机体系结构、处理器设计、数字信号处理.