分　数：	
评卷人：	

華中科技大學

研究生（数据中心技术）课程论文（报告）

题　目：**NeuGraph:** 大图的并行深度神经网络计算

学　　号　M201973014

姓　　名　何雨迪

专　　业　计算机技术

课程指导教师　施展、曾令仿

院（系、所）　武汉光电国家研究中心

2019 年 **11** 月 **5** 日

NeuGraph: 大图的并行深度神经网络计算

国光硕 1905 班　M201973014　何雨迪

1 论文背景

众所周知，图是真实世界中许多数据的抽象表示，例如网页链接图、社交网络图、交通网络图和知识图谱等，由于图中蕴含了丰富的具有潜在价值的信息，对图的研究和一直成为近些年的热点。在大数据时代下，实际的图数据很容易扩展到数百万节点和边，因此，如何设计一个性能高、扩展性好的图处理模型成为关键的问题。

近十年，深度学习在语音、图像、自然语言处理等方面展示出顶尖的性能，其对这类数据结构的特征提取在业界一度广受好评。但是，现实中普遍存在的图却是一个难题。与图像和文本不同，图的结构是不规则的，使得一些深度学习算法无法应用于图；另一方面，现实世界中的图普遍是动态变化的，这就需要不同的模型架构来解决特定的问题。

为了解决这些挑战，相关研究学者付出了大量努力，提出了图神经网络（GNN）将图和神经网络结合，利用图的结构传递点和边的信息，通过深度神经网络计算点和边的信息，在节点分类、链接预测等方面有广泛的应用。随之而来又出现了一个问题，无论是现有的深度学习框架还是现有的图系统都不能很好地支持 GNN 算法，极大了限制了 GNN 的研究和发展，所以作者提出了 NeuGraph 框架——基于数据流的深度学习系统的 GNN 处理框架，桥接了深度学习框架和图处理系统，使得在 DL 框架上对图执行高效可扩展的深度学习算法成为可能。

2 研究现状

2005 年 Gori 等人（Gori et al 2005）最先提出图神经网络的概念，并由 Scarselli 等人（Scarselli et al 2009）进一步阐明。这些早期的研究以迭代的方式通过循环神经架构传播邻近信息来学习目标节点的表示，直到达到稳定的固定点。

2014 年 Bruna 等人（Bruna et al 2014）提出了关于图卷积网络的第一项重要研究，他们基于谱图论开发了一种图卷积的变体。自此，基于谱的图卷积网络不断改进、拓展、进阶，基于空间的图卷积网络开始快速发展。这些方法通过聚集

近邻节点的信息，直接在图结构上执行卷积。

Mikolov 等人(Mikolov et al 2013) 提出图嵌入，学习用低维向量表示图节点、边或子图。由于传统的机器学习方法通常依赖于手工工程特征，其灵活性和成本都受到限制，图嵌入方法既可以最大程度得保留图结构信息，又简化了图的处理。

Perozzi 等人(Perozzi et al 2014)提出 DeepWalk，DeepWalk 是第一个以无监督学习的节点嵌入算法，将 skip gram 模型应用于随机游动，它在训练过程中类似于词嵌入，不仅能表示节点，还能表示出节点之间的拓扑关系。

2017 年 Gilmer 等人(Gilmer et al 2017)提出了一种基于图的有监督学习的通用框架，称为消息传递神经网络（MPNNS）。MPNN 框架抽象了几种最流行的图形结构数据模型之间的共同点，如图卷积中的门控图神经网络、交互网络等。

2018 年 Velickovic 等人（Velickovic et al 2018）提出了一种图注意网络（GAT），它将注意机制引入传播步骤中，通过对每个节点的邻居进行关注来计算每个节点的隐藏状态，遵循自关注策略。

3 提出的 NeuGraph 框架

NeuGraph 结合了数据流抽象和顶点编程抽象产生了一个全新的编程模型——SAGA-NN 模型。与传统的用户定义函数（UDF）表示顶点程序不同，SAGA-NN 中的用户定义函数在顶点或边数据上做神经网络计算，同时不必担心底层系统的实现（例如 GPU 内存管理或调度）。

GPU 的使用可以大大提高图处理的效率，但现实世界中的大图难以整个存储在 GPU 中，NeuGraph 提出了 2D 图分割策略，将点和边划分为子块，将这样的点和边的子块输入到 GPU 中抽象成一个子数据流图。

为了进一步扩展性能，NeuGraph 采用了一种新的拓扑感知调度策略，针对现代多 GPU 系统提出了基于链式的并行流处理，使在多 GPU 系统上执行图处理的效率大大提升。

3.1 SAGA-NN 模型

SAGA-NN 模型的全称是 Scatter-ApplyEdge-Gather-ApplyVertex with Neural Networks。顾名思义，它包含 4 个阶段 Scatter，ApplyEdge，Gather，ApplyVertex。

NeuGraph 将 GNN 建模为一系列的 SAGA 阶段。第 1 阶段是 Scatter，开始每个顶点都有一个特征向量，然后沿着边将顶点数据传递到其相邻边上以构造边向量，边的数据形式以源点，终点和边数据组成；第 2 阶段是 ApplyEdge，将构造的边向量输入到用户定义的神经网络中做计算，输出 edge output 向量；第 3 阶段是 Gather，将上一阶段产生的同一终点的边输出向量聚集起来，输出一个累积向量；第 4 阶段是 ApplyVertex，输入每个顶点的累积向量，执行用户定义的神经网络计算，以生成下一层对应顶点的特征。

ApplyVertex 和 ApplyEdge 是 SAGA-NN 模型提供的两个用户定义的神经网络，分别是对点和边做神经网络计算；Scatter 和 Gather 这两个阶段执行数据传播并准备数据收集，以作为输入送入 ApplyEdge 和 ApplyVertex，它们是由系统隐式触发执行。

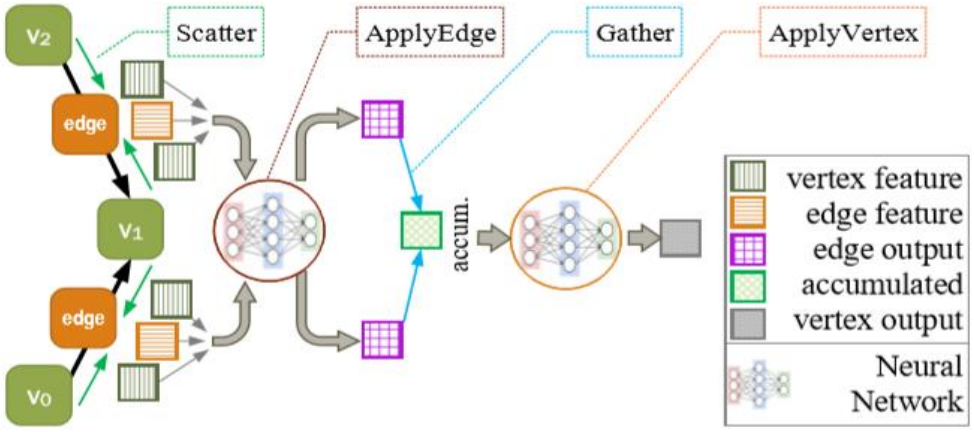

Figure 2: SAGA-NN stages for each layer of GNN.

3.2 基于块的图分割

与 DNNs 一样，GPU 的有效使用对 GNNs 的性能至关重要，特别是对于大型图。然而现有的 DL 框架不能直接处理 GPU 上的大图，因为图数据不能完全存储在 GPU 存储器中。为了实现超出 GPU 内存物理限制的可伸缩性，NeuGraph 在数据流抽象之上引入了基于块的图分割——2D 图分割。

如图 4 所示，2D 图分割策略将节点随机地分割成 P（P=2）个大小相等的不相交顶点块，边分为 P*P 组，并平铺为矩阵的形式，边块 Ei,j 包含第 i 个顶点块指向第 j 个顶点块的边。通过将图数据分割成块，NeuGraph 可以逐个处理边块，

并且只需输入与边块相关联的源顶点块和目标顶点块即可。

对于前馈计算计算，NeuGraph 为每个目标顶点分区转换一个数据流子图，如对图 4 中边块的第一列生成一个数据流子图，产生对 0 和 3 号目标顶点的更新。

这里有必要讨论一下，为何生成数据流子图时采用面向列的方式而非行？源顶点决定边块的行，目标顶点决定边块的列，对于前馈传播计算，数据从源顶点流向目标顶点。使用面向行的处理失去了重复使用累积的顶点数据块的机会，这些顶点数据块的总大小可能大于 GPU 内存的大小，这就带来了冗余的数据迁移操作。

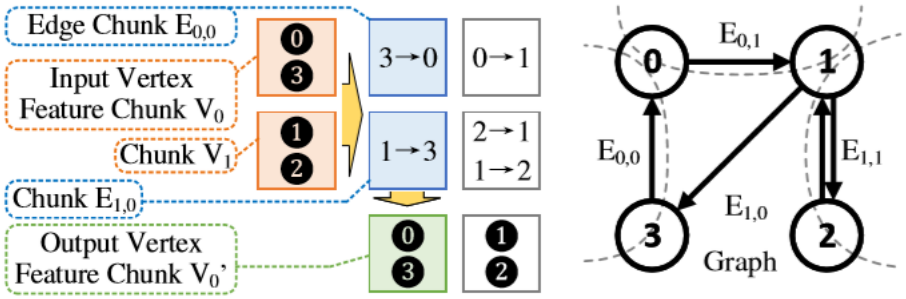

Figure 4: 2D Partitioning of a graph, here $P = 2$.

3.3 多 GPU 上的链式并行流处理

为了进一步提高可扩展性，NeuGraph 通过在多个 GPU 上划分基于块的数据流来并行化训练。因为数据流图具有并行性，很容易并行化，可以分配多个 GPU 用于协同处理的不同列的数据流子图。

图 7 显示了一个典型的 8-GPU 服务器的拓扑结构，其中 GPU 通过多级 PCIE 接口连接到 CPU/DRAM（主机内存）。当同时从主机存储器读取边/顶点数据时，由多个通信路径共享的上层链路很容易成为瓶颈。例如 GPU0 和 1 连接到主机时共享了一个 PCIe 接口，当传输数据时，它们都只能达到其峰值带宽的一半。

为了最大化 GNN 在多个 GPU 上的并行度，防止共享的链路成为性能瓶颈，NeuGraph 采用基于链的流调度方案，让 GPU 传输顶点块给它相邻的 GPU，避免共享链路的另一个 GPU 向主机索要数据而造成拥塞。

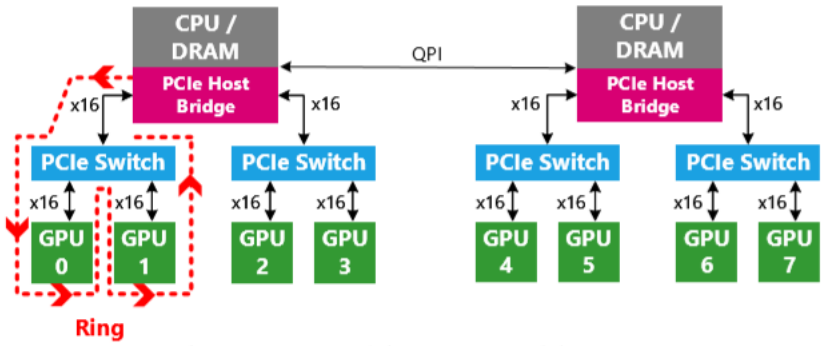

Figure 7: Multi-GPU architecture

基于链的流式调度策略下，通常 GPU 需要执行以下 2 个操作：1）从主机内存加载点和边块，或者从链路中前一个 GPU 的设备内存加载数据块，2）执行本地计算。NeuGraph 为了更好地重叠这两个操作采用了一个协调的调度，如图 8 所示。

根据互连拓扑将 GPU 分组为多个虚拟 GPU，例如 GPU0 和 GPU1 构成一个虚拟 GPU，GPU2 和 GPU3 组成另一个虚拟 GPU。最开始，GPU0 和 GPU2 中从主机内存中导入顶点块 V0，导入成功后，GPU0 和 GPU2 开始对顶点块 V0 做计算，接下来从主机内存中导入顶点块 V1 的同时，将顶点块 V0 传输给 GPU1 和 GPU3；下一步 GPU1 和 GPU3 在本地处理数据块 V0 之后删除它，因为所有虚拟 GPU 都已经使用了该块，整个过程以这样的流水线方式继续，直到所有顶点数据块都被加载和处理。

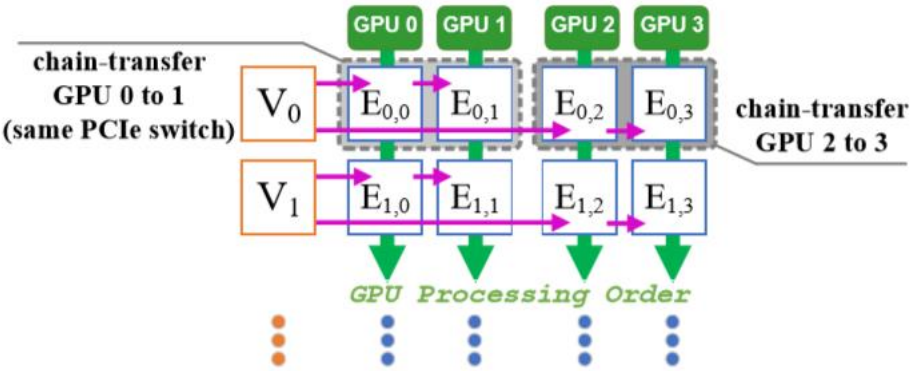

Figure 8: NeuGraph transfers vertex chunks along the chain.

4 评估

为了评估 NeuGraph 的性能，论文通过在多个 GNN 和数据集上对 Neugraph 进行评估，最终证实了它的高效率和可扩展性。

图 13 展示了 DGL、TensorFlow、TF-SAGA 和 Neugraph 这几个系统在小数据集上的性能比较。总体而言，Neugraph 对比 TensorFlow 有 2.5 倍的平均加速比（高达 5.0 倍），与 DGL 相比平均加速 8.1 倍（高达 19.2 倍）。

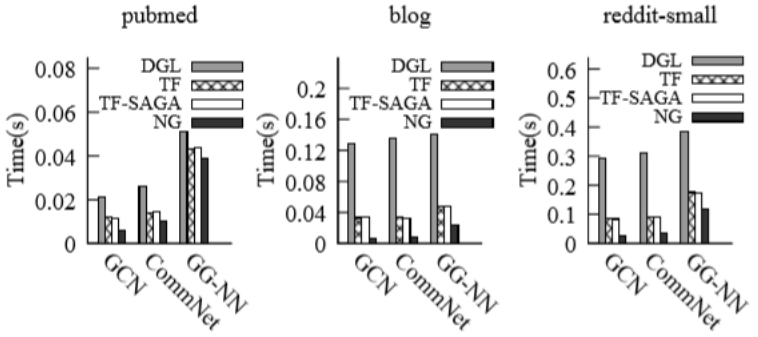

Figure 13: End-to-end performance comparison among DGL, TensorFlow (TF), TF-SAGA and NeuGraph (NG) on small datasets. GraphSAGE runs OOM.

图 14 显示了 NeuGraph 和其他模型在大数据集上的端到端的比较结果，Neugraph 实现了比 Tensorflow CPU 大 16 47 倍的加速比，而 Tensorflow CPU 是目前大型图形的解决方案；在不同的模型和数据集上，Neugraph 的速度比启用 GPU 的 TF-SAGA 快 2.4~4.9 倍。

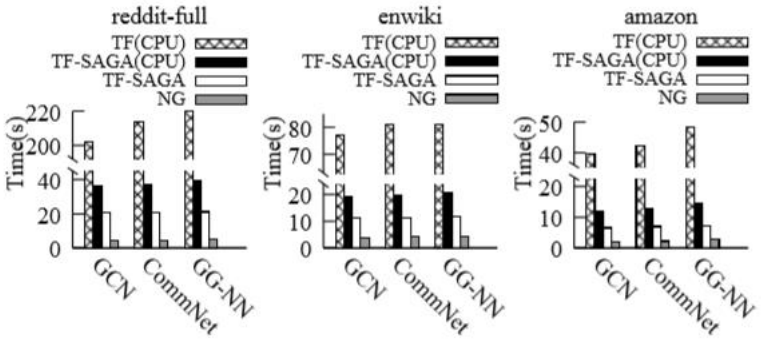

Figure 14: NeuGraph end-to-end performance comparisons on different large datasets. TensorFlow uses CPU-only mode as OOM occurs on GPU. TF-SAGA (CPU) is configured to run on CPU only, whereas TF-SAGA is GPU-enabled.

5 总结

这篇文章的主要工作是提出了一个全新的处理图神经网络的系统 NeuGraph，NeuGraph 实现的 SAGA-NN 编程框架允许用户自定义神经网络对图中的边和点执行操作而无需关注底层实现，同时 NeuGraph 将图相关的优化策略（如图分割、调度和并行化管理）融入到深度学习框架中。最终不管是在大图还是小图处理上，都能达到优于目前系统的高效率和高扩展性。

6 参考文献

【1】 Zhou J , Cui G , Zhang Z , et al. Graph Neural Networks: A Review of Methods and Applications[J]. 2018.

【2】 F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Mon-fardini. Computational capabilities of graph neural networks.IEEE Transactions on Neural Networks, 20(1):81–102, 2009

【3】 J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun. Spectral networksand locally connected networks on graphs.international conferenceon learning representations, 2014.

【4】 T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient esti-mation of word representations in vector space.arXiv preprintarXiv:1301.3781, 2013

【5】 B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learningof social representations. InProceedings of the 20th ACM SIGKDDinternational conference on Knowledge discovery and data mining,pages 701–710. ACM, 2014.

【6】 J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl.Neural message passing for quantum chemistry.arXiv preprintarXiv:1704.01212, 2017.

【7】 P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, andY. Bengio. Graph attention networks.international conference onlearning representations, 2018.