**专利申请明细表**

**兹全权委托北京博思佳知识产权代理有限公司办理下列申请专利有关事项**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 发 明 名 称 | | 一种高性能图神经网络硬件加速器 | | |
| 第一  申请人 | 姓名或名 称 | 之江实验室 | | |
| 地址、邮编 | 默认：浙江省杭州市余杭区之江实验室南湖总部，311121  日本专利：之江实验室 中国310023浙江省杭州市余杭区文一西路1818号 | | |
| 企业机构代码或身份证号 | 12330000MB1478604D | | |
| 第二  申请人  （如果为多个申请单位共同申请，可填） | 姓名或名 称 | 华中科技大学 | | |
| 地址、邮编 | 地址：湖北省武汉市洪山区珞喻路1037号，430074 | | |
| 企业机构代码或身份证号 | 12100000441626842D | | |
| 发明人  或设计人 | 姓 名 | 张宇，黄浚，赵进，余辉，张湛 | 第一发明人身份证号 | 431122198707170515 |
| 专利负责人 | 姓名 | 吴梅英，18662590857，wumeiying@zhejianglab.com  地址：浙江省杭州市余杭区之江实验室南湖总部主楼1402办公室，311100 | | |
| 地址、邮编 |
| 电话/E-mail |
| 技术（撰写）  联系人 | 姓名 | 黄浚，13459307227，jun\_huang@hust.edu.cn  地址：浙江省杭州市余杭区之江实验室南湖总部11号楼B座B102室，311121 | | |
| 地址、邮编 |
| 电话/E-mail |
| 申请专利类型 | | 🗹发明　　　　　实用新型　　　　外观设计 | | |
| 催缴费联系人电话、Email、地址 | | 吴梅英，18662590857，wumeiying@zhejianglab.com，地址：浙江省杭州市余杭区之江实验室南湖总部主楼1402办公室，311100 | | |

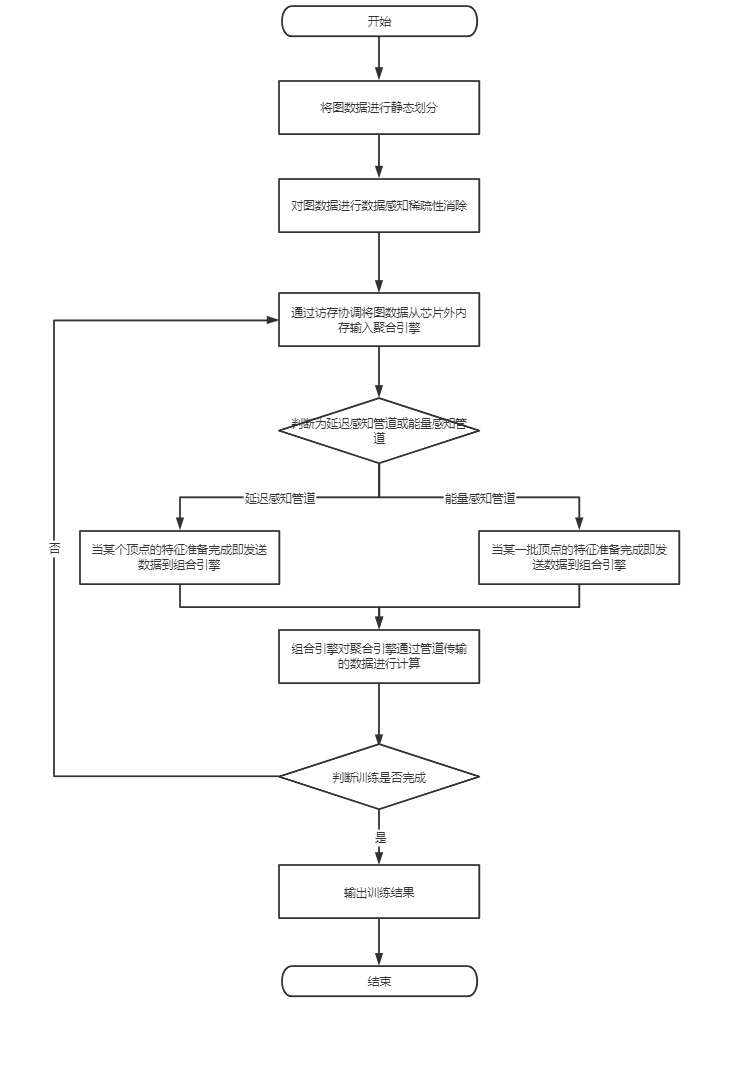
注：

1、红色字体部分需要填写或选择

说 明 书 摘 要

本发明实现了一种特定于图卷积神经网络的加速器，该加速器主要对图卷积神经网络的聚合阶段和组合阶段进行加速。在聚合阶段，由于图处理是一个动态且不规则的执行模式，因此提出了聚合引擎来对聚合阶段的不规则性进行缓解，同时在聚合阶段添加一个缓冲区缓存度数高于阈值的顶点，利用图数据结构的空间局部性，降低访存未命中率；在组合阶段的执行模式是与神经网络类似，静态且有规则的因此提出了组合引擎来利用组合阶段的规则性，从而实现在聚合阶段和组合阶段的加速。此外，本发明提出的这两种引擎还能够利用各种并行性，有效地重用高度可重用数据。本发明还通过引擎间流水线优化整个系统，实现相间融合和基于优先级的芯片外内存访问协调来提高芯片外带宽利用率。

摘 要 附 图



权 利 要 求 书

1. 一种高性能图卷积神经网络的加速器，其架构主要分为以下几个部分：
2. 聚合引擎。引入了一种顶点分散的处理模式，采用静态图划分来增强数据重用，采用动态稀疏性消除技术减少不必要的数据访问，添加一个缓冲区缓存度数高于阈值的顶点来提高访存命中率。
3. 组合引擎。采用收缩阵列来处理聚合引擎中传入的数据，并通过使用颗粒的大小将其分为独立工作模式和合作工作模式。
4. 聚合组合缓冲区。为了重用聚合引擎产生的聚合结果，在两个引擎间添加了聚合组合缓冲区，该缓冲区可由聚合引擎写入，也可由组合引擎读取。
5. 芯片外内存的访问协调。由于聚合引擎和组合引擎的实际工作负载通常不同，因此为其各分配一个芯片外内存的方法会造成带宽浪费。而两个引擎在运行时同时访问内存，导致访问位置的频繁切换，因此预定义了一个优先级来进行访问。
6. 如权利要求1所述的方法，其特征在于，所述步骤（1）包括如

下子步骤：

（1.1）图划分（静态）。在首轮迭代计算之前，先使用顶点区间和边碎片的抽象来划分图数据，直接将将压缩稀疏列数据格式作为输入，将顶点组织为若干个间隔，边组织成若干个碎片，这些间隔和碎片是不相交的。碎片高度由输入缓冲器的容量决定，而碎片宽度由聚合缓冲器的容量决定。边缘缓冲区的大小会影响高度和宽度，因为它可以容纳每个碎片的所有边缘。

（1.2）数据感知稀疏性消除（动态）。通过数据重用优化，由于图连接是稀疏分布的，提出了一种基于窗口的滑动和收缩方法。窗口滑动，对于每个顶点间隔，顶部的碎片窗口逐渐向下滑动。它将不会停止，直到一个边缘出现在它的顶部一行。然后创建一个大小相同的新窗口，其顶部行跟随其上一个窗口的底部行。这样，窗口就会不断地升起，向下滑动，然后停止。所有窗口停止的位置都被记录为有效的碎片。窗口收缩，虽然窗口滑动可以捕获最有效的边缘，但稀疏性仍然存在于底部，这是因为上述的滑动方向是向下的。为了减少这部分的稀疏性，在这里缩小窗口。具体来说，每个记录的窗口的底部一行向上移动，直到它遇到一个边缘，然后窗口缩小。与之前的分区不同，最终碎片的大小通常由于窗口的缩小而不同。

（1.3）顶点分散处理模式。单指令多数据流核具有两种并行处理边缘的方式。本发明采用的是顶点分散处理模式。它将每个顶点的特征向量内的元素聚合分配给所有的单指令多数据流核，如果一个顶点无法占据所有的核，自由的核可以被分配给其他顶点。因此所有的核心总是被占用，不存在工作负载不平衡。此外，由于利用了顶点内并行性，因此单个顶点的顶点延迟要小于同时处理多个顶点。

（1.4）添加一个缓冲区。在聚合阶段添加缓冲区，预先将一些顶点度数高于阈值的顶点预取到缓冲区中，阈值设定由图大小所决定。在进行预取时也能够进行图划分（静态），从而提高并行率，且图数据访问时大多数情况下均访问高度数顶点，因此利用了图数据访问的空间局部性，提高访存命中率。

（1.5）在聚合阶段完成处理后将处理结果传入流水线中，以便组合引擎读取数据。

1. 如权利要求1所述的方法，其特征在于，所述步骤（2）包括如

下子步骤：

（2.1）组合引擎主要是由收缩阵列组成，为了适应聚合引擎的两种处理模式，组合引擎集成了多个阵列，允许多颗粒地使用这些收缩模块，因此组合引擎分为独立的工作模式和合作的工作模式。

（2.2）独立的工作模式。在这种模式下，收缩期模块相互独立地工作。每个模块都处理一小组顶点的矩阵乘法操作，每个模块的权重参数都可以直接从权重缓冲区中访问，并仅在模块中重用。这种模式的优点是顶点延迟较低，因为我们可以在这一小群顶点的聚合特性准备好后立即处理它们的组合操作，而无需等待更多的顶点。

（2.3）合作的工作模式。这些收缩期模块还可以进一步组装在一起，同时处理更多的顶点。这种模式需要在执行组合操作之前将大量顶点的聚合特征组合在一起。其优点是，重量参数可以从重量缓冲器流到下游收缩模块，然后逐渐流到上游模块，所有收缩阵列都可以重复使用。

（2.4）在每次组合引擎完成计算后判断该次迭代是否为最后一次迭。若为最后一次迭代则将结果进行输出，结束图卷积神经网络；若部位最后一次迭代，将中间结果传回缓冲区，由聚合阶段重新提取，回到（1.2）。

1. 如权利要求1所述的方法，其特征在于，所述步骤（3）包括如

下子步骤：

（3.1）延迟感知流水线。在此流水线模式下，组合引擎在收缩期模块独立模式下工作。聚合特性在聚合引擎中逐顶点生成，一旦一小群顶点的聚合特性准备好，将立即处理以下组合。因此，每个顶点的平均处理延迟可以更低。

（3.2）能量感知流水线。能量感知流水线在组合发动机中使用收缩期模块协同模式。逐顶点的处理将更改为突发模式，即每次都会一起处理大量的顶点。虽然顶点延迟较长，但由于在合并的收缩期阵列中没有冗余访问的权重传播，可以减少能量消耗。

1. 如权利要求1所述的方法，其特征在于，所述步骤（4）包括如

下子步骤：

（4.1）预先定义了一个访问优先级（边缘>输入特征>权重> 输出特征）来组装不连续请求，访问请求被逐批执行。因此，当前批中的低优先级访问是在下一批中的高优先级访问之前处理的，而不是总是首先进行高优先级访问。随着连续性的提高，行缓冲区的利用率可以显著提高。接下来，我们重新映射这些重新排序的地址，以使用低位来索引通道和块。从而进一步利用内存通道级和块级的并行性。

说 明 书

一种高性能图神经网络硬件加速器

**技术领域**

本发明属于机器学习的图神经网络领域，具体涉及一种特定于图卷积神经网络的加速器设计，利用图卷积神经网络不同阶段的特征来进行优化加速。

**背景技术**

受神经网络强大学习能力的启发，图卷积神经网络被提出作为标识和处理图数据的有效模型。图卷积神经网络将图数据转化为一个低维空间，同时最大化保持结构和属性信息，然后为之后的训练和推理构建神经网络。最近，图卷积神经网络吸引了工业界和学术界的大量努力来解决节点分类、链路预测、图聚类和推荐等问题。因此，图卷积神经网络逐渐成为数据中心的一个新的工作量家庭成员，如谷歌、脸书和阿里巴巴。

卷积层有两个主要的执行阶段：聚合和组合。聚合阶段中的行为大多数为图处理行为。它在很大程度上依赖于固有的随机和稀疏的图结构。处理每个顶点需要从其所有的源邻居中聚合特征。而这些源邻居的数量和位置在不同顶点之间存在显著差异。因此，在每个顶点的聚合阶段，计算图和内存访问模式是动态且无规则的。组合阶段更像神经网络。它使用多层感知器将每个顶点的特征向量转换为一个新的感知器，通常使用矩阵向量乘法表示。由于神经网络层内每个神经元的连接模式相同，因此每个顶点的组合阶段你的计算图和内存访问模式是静态且有规则的。此外，在这两个阶段中，图卷积神经网络与传统的图分析相比还有一些特性。首先，在传统的图分析中，顶点的特征较短且固定，而在图卷积神经网络中，每个顶点的特征向量较长且跨层可变，这在聚合阶段引入了高度顶点内并行性。其次，传统的基于多层感知器的神经网络的参数从不共享，而这些参数可以在图卷积神经网络中的顶点之间共享。这在导致组合阶段产生了大量可重用的顶点间数据。第三，这两个阶段交替执行，在各个阶段之间存在一个固有的数据流，提供了一个将这些阶段融合执行的机会。

基于上述原因，本发明提出了一种特定于图卷积神经网络的加速器，使用一个混合架构来执行图卷积神经网络，利用该领域中的各种并行性。

**发明内容**

针对现有的条件，本发明提出一种特定于图卷积神经网络的加速器，解决了图卷积神经网络聚合阶段的不规则性并利用组合阶段的规则性，在聚合阶段添加一个缓冲区缓存度数高于阈值的顶点，利用图数据结构的空间局部性，降低访存未命中率，同时提出了聚合引擎和组合引擎之间的流水线，有效地融合了聚合阶段和组合阶段地执行，提升了系统的性能。

为实现上述目的，本发明包括以下步骤：

（1）聚合引擎。引入了一种顶点分散的处理模式，图划分（静态），数据感知稀疏性消除（动态）以及添加一个缓冲区缓存度数高于阈值的顶点来提高访存命中率。顶点分散的处理模式将每个顶点的顶点特征向量内的元素聚合分配给所有的核，如果一个顶点不能占据所有的核，自由的核可以被分配给其他的顶点。因此，所有的核心总是处于工作状态，没有工作负载不平衡。此外，由于利用了顶点内并行性，因此单个顶点的顶点延迟要小于同时处理多个顶点。图划分（静态）利用顶点区间和边碎片地抽象来划分图数据，为数据感知稀疏性消除打好基础。直接使用压缩稀疏列的数据格式作为输入，将顶点分为若干个间隔，将边组织成若干个碎片，这些间隔和碎片是不相交的。在实践中，边碎片通常不是正方形的，碎片高度由输入缓冲器的容量决定，而碎片宽度由聚合缓冲器的容量决定。边缓冲区的大小会影响高度和宽度，因为它可以容纳每个碎片的所有边。数据感知稀疏性消除（动态）提出了一种基于窗口的滑动和收缩方法。添加一个缓冲区缓存度数高于阈值的顶点，从而利用图数据访问的空间局部性提高访存命中率。

（2）组合引擎。每个顶点上的组合操作就像一个神经网络，其执行是常规的，但计算密集型的。本发明基于收缩阵列设计了组合引擎，能在独立工作模式和合作工作模式下工作。在独立工作模式种，每个收缩模块独立工作，只处理一小组顶点的矩阵乘法操作，每个模块的权重参数都可以直接从权重缓冲区中访问，并仅在模块中重用，这种模式的优点是顶点延迟较低。在合作工作模式中，收缩模块还可以进一步组装在一起，同时处理更多的顶点，权重参数可以从权重缓冲器流到下游收缩模块，然后逐渐流到上游模块，所有收缩阵列都可以重复使用，这有助于减少能源消耗。

（3）聚合组合缓冲区。了重用聚合引擎产生的聚合结果，我们在两个引擎之间添加了一个聚合组合缓冲区。此缓冲区可以由聚合引擎写入，也可以由组合引擎读取。通过这种方式，聚合和组合的执行就被解耦了，从而实现了一个引擎间的流水线。

为了适应不同应用程序的需求，本发明提供了以下两种流水线模式。

延迟感知流水线。在此流水线模式下，组合引擎在收缩模块独立模式下工作。聚合特性在聚合引擎中逐顶点生成，一旦一小群顶点的聚合特性准备好，将立即处理以下组合。因此，每个顶点的平均处理延迟可以更低。

能量感知流水线。能量感知流水线在组合发动机中使用收缩模块合作模式。逐顶点的处理将更改为突发模式，即每次都会一起处理大量的顶点。虽然顶点延迟较长，但由于在合并的收缩期阵列中没有冗余访问的权重传播，可以减少能量消耗

（4）芯片外内存访问的协调。由于聚合和组合引擎之间的实际工作负载通常不同，因此很难确定两个引擎之间的内存带宽比，内存系统的分离将增加配置开销，并造成带宽浪费，因此本发明只使用一个芯片外内存。两个引擎在运行时访问这个内存，导致访问位置的频繁切换，导致效率低下。总共有四个缓冲区（聚合引擎中的边缓冲区和输入缓冲区，以及组合引擎中的权重缓冲区和输出缓冲区）将用于访问芯片外内存。由于间隔处理和流水线机制，这些访问通常同时进行，如果我们依次处理这些访问请求，不连续地址将极大地降低动态随机存取内存中行缓冲区的利用率。本发明预先定义了一个访问优先级（边缘>输入特征>权重> 输出特征）来组装不连续请求，访问请求被逐批执行。随着连续性的提高，行缓冲区的利用率可以显著提高。

上述特定于图卷积神经网络的加速器，实施步骤如下：

（1）聚合引擎。引入了一种顶点分散的处理模式，采用静态图划分来增强数据重用，采用动态稀疏性消除技术减少不必要的数据访问，添加一个缓冲区缓存度数高于阈值的顶点来提高访存命中率。其主要包括如下子步骤：

（1.1）图划分（静态）。在首轮迭代计算之前，先使用顶点区间和边碎片的抽象来划分图数据，直接将将压缩稀疏列（压缩稀疏列）数据格式作为输入，将顶点组织为若干个间隔，边组织成若干个碎片，这些间隔和碎片是不相交的。碎片高度由输入缓冲器的容量决定，而碎片宽度由聚合缓冲器的容量决定。边缘缓冲区的大小会影响高度和宽度，因为它可以容纳每个碎片的所有边缘。

（1.2）数据感知稀疏性消除（动态）。通过数据重用优化，由于图连接是稀疏分布的，提出了一种基于窗口的滑动和收缩方法。窗口滑动，对于每个顶点间隔，顶部的碎片窗口逐渐向下滑动。它将不会停止，直到一个边缘出现在它的顶部一行。然后创建一个大小相同的新窗口，其顶部行跟随其上一个窗口的底部行。这样，窗口就会不断地升起，向下滑动，然后停止。所有窗口停止的位置都被记录为有效的碎片。窗口收缩，虽然窗口滑动可以捕获最有效的边缘，但稀疏性仍然存在于底部，这是因为上述的滑动方向是向下的。为了减少这部分的稀疏性，在这里缩小窗口。具体来说，每个记录的窗口的底部一行向上移动，直到它遇到一个边缘，然后窗口缩小。与之前的分区不同，最终碎片的大小通常由于窗口的缩小而不同。

（1.3）顶点分散处理模式。单指令多数据流核具有两种并行处理边缘的方式。本发明采用的是顶点分散处理模式。它将每个顶点的特征向量内的元素聚合分配给所有的单指令多数据流核，如果一个顶点无法占据所有的核，自由的核可以被分配给其他顶点。因此所有的核心总是被占用，不存在工作负载不平衡。此外，由于利用了顶点内并行性，因此单个顶点的顶点延迟要小于同时处理多个顶点。

（1.4）添加一个缓冲区。在聚合阶段添加缓冲区，预先将一些顶点度数高于阈值的顶点预取到缓冲区中，阈值设定由图大小所决定。在进行预取时也能够进行图划分（静态），从而提高并行率，且图数据访问时大多数情况下均访问高度数顶点，因此利用了图数据访问的空间局部性，提高访存命中率。

（1.5）在聚合阶段完成处理后将处理结果传入流水线，以便组合引擎读取数据。

（2）组合引擎。采用收缩阵列来处理聚合引擎中传入的数据，并通过使用颗粒的大小将其分为独立工作模式和合作工作模式。其主要包括如下子步骤：

（2.1）组合引擎主要是由收缩阵列组成，为了适应聚合引擎的两种处理模式，组合引擎集成了多个阵列，允许多颗粒地使用这些收缩模块，因此组合引擎分为独立的工作模式和合作的工作模式。

（2.2）独立的工作模式。在这种模式下，收缩期模块相互独立地工作。每个模块都处理一小组顶点的矩阵乘法操作，每个模块的权重参数都可以直接从权重缓冲区中访问，并仅在模块中重用。这种模式的优点是顶点延迟较低，因为我们可以在这一小群顶点的聚合特性准备好后立即处理它们的组合操作，而无需等待更多的顶点。

（2.3）合作的工作模式。这些收缩期模块还可以进一步组装在一起，同时处理更多的顶点。这种模式需要在执行组合操作之前将大量顶点的聚合特征组合在一起。其优点是，重量参数可以从重量缓冲器流到下游收缩模块，然后逐渐流到上游模块，所有收缩阵列都可以重复使用。

（2.4）在每次组合引擎完成计算后判断该次迭代是否为最后一次迭。若为最后一次迭代则将结果进行输出，结束图卷积神经网络；若部位最后一次迭代，将中间结果传回缓冲区，由聚合阶段重新提取，回到（1.2）。

（3）聚合组合缓冲区。为了重用聚合引擎产生的聚合结果，在两个引擎间添加了聚合组合缓冲区，该缓冲区可由聚合引擎写入，也可由组合引擎读取。其主要包括如下子步骤：

（3.1）延迟感知流水线。在此流水线模式下，组合引擎在收缩期模块独立模式下工作。聚合特性在聚合引擎中逐顶点生成，一旦一小群顶点的聚合特性准备好，将立即处理以下组合。因此，每个顶点的平均处理延迟可以更低。

（3.2）能量感知流水线。能量感知流水线在组合发动机中使用收缩期模块协同模式。逐顶点的处理将更改为突发模式，即每次都会一起处理大量的顶点。虽然顶点延迟较长，但由于在合并的收缩期阵列中没有冗余访问的权重传播，可以减少能量消耗。

（4）芯片外内存的访问协调。由于聚合引擎和组合引擎的实际工作负载通常不同，因此为其各分配一个芯片外内存的方法会造成带宽浪费。而两个引擎在运行时同时访问内存，导致访问位置的频繁切换，因此预定义了一个优先级来进行访问。其主要包括如下子步骤：

（4.1）预先定义了一个访问优先级（边缘>输入特征>权重> 输出特征）来组装不连续请求，访问请求被逐批执行。因此，当前批中的低优先级访问是在下一批中的高优先级访问之前处理的，而不是总是首先进行高优先级访问。随着连续性的提高，行缓冲区的利用率可以显著提高。接下来，我们重新映射这些重新排序的地址，以使用低位来索引通道和块。从而进一步利用内存通道级和块级的并行性。

总体而言，通过本发明所构思的以上技术方案与现有技术相比，能够取得下列有益效果：

1. 聚合阶段并行性高，数据重用高，不必要的数据访问少：通过采用了顶点分散的处理模式，使每个单指令多数据流核总处于工作状态，有效利用了顶点内的并行性。通过图划分（静态），将可重用的图数据进行划分，使其中间聚合结果能够保留在缓冲区中，提高数据重用的效率。通过数据感知稀疏性消除，使用基于窗口的滑动和收缩方法，将每次范围的范围缩小，消除了大量冗余访问。
2. 组合阶段数据重用高：通过使用收缩阵列来设计组合引擎，通过不同的组合方式设计独立和合作工作模式。在这两种工作模式下，所执行的矩阵乘法操作都能够共享权重参数，提高了数据重用。
3. 能够适应多种程序需求：本发明在聚合引擎和组合引擎之间加入了缓冲区，提供了延迟感知流水线和能量感知流水线两种模式。若程序需要以最快的速度完成图卷积神经网络，则使用延迟感知流水线；若程序需要以尽可能少的能量消耗来完成图卷积神经网络，则使用能量感知流水线。
4. 芯片外内存的访问效率高：本发明通过预定义访问优先级来组装不连续请求，访问请求逐批执行。随着连续性提高，行缓冲区利用率能得到显著提高。之后重新映射这些重新排序的地址，使用低位来索引通道和块，进一步利用内存通道级和块级的并行性。

**附图说明**

图1为本发明聚合阶段的流程图。

图2为本发明组合阶段的流程图。

图3为本发明芯片外内存访问协调的流程图。

**实施方式**

为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图，对本发明进行进一步详细说明。此外，下面所描述的本发明的各个实施方式中所涉及到的技术特征只要彼此之间未构成冲突就可以相互组合。

本发明提供的特定于图卷积神经网络的加速器，其主要思想为：将图卷积神经网络聚合阶段和组合阶段融合执行利用聚合阶段的高顶点内并行性和组合阶段的高数据重用，同时在聚合阶段添加一个缓冲区缓存度数高于阈值的顶点，利用图数据结构的空间局部性，降低访存未命中率，并对两个阶段的内存访问进行优先级排序，大大加快了图卷积神经网络网络收敛速度，减少了计算冗余，降低了缓存缺失率，提高了内存访问效率。

本发明提供的特定于图卷积神经网络的加速器，其流程如图1和图2所示，包括聚合阶段，组合阶段。具体如下：

（1）聚合引擎。引入了一种顶点分散的处理模式，采用静态图划分来增强数据重用，采用动态稀疏性消除技术减少不必要的数据访问，添加一个缓冲区缓存度数高于阈值的顶点来提高访存命中率。其主要包括如下子步骤：

（1.1）图划分（静态）。在首轮迭代计算之前，先使用顶点区间和边碎片的抽象来划分图数据，直接将将压缩稀疏列（压缩稀疏列）数据格式作为输入，将顶点组织为若干个间隔，边组织成若干个碎片，这些间隔和碎片是不相交的。碎片高度由输入缓冲器的容量决定，而碎片宽度由聚合缓冲器的容量决定。边缘缓冲区的大小会影响高度和宽度，因为它可以容纳每个碎片的所有边缘。

（1.2）数据感知稀疏性消除（动态）。通过数据重用优化，由于图连接是稀疏分布的，提出了一种基于窗口的滑动和收缩方法。窗口滑动，对于每个顶点间隔，顶部的碎片窗口逐渐向下滑动。它将不会停止，直到一个边缘出现在它的顶部一行。然后创建一个大小相同的新窗口，其顶部行跟随其上一个窗口的底部行。这样，窗口就会不断地升起，向下滑动，然后停止。所有窗口停止的位置都被记录为有效的碎片。窗口收缩，虽然窗口滑动可以捕获最有效的边缘，但稀疏性仍然存在于底部，这是因为上述的滑动方向是向下的。为了减少这部分的稀疏性，在这里缩小窗口。具体来说，每个记录的窗口的底部一行向上移动，直到它遇到一个边缘，然后窗口缩小。与之前的分区不同，最终碎片的大小通常由于窗口的缩小而不同。

（1.3）顶点分散处理模式。单指令多数据流核具有两种并行处理边缘的方式。本发明采用的是顶点分散处理模式。它将每个顶点的特征向量内的元素聚合分配给所有的单指令多数据流核，如果一个顶点无法占据所有的核，自由的核可以被分配给其他顶点。因此所有的核心总是被占用，不存在工作负载不平衡。此外，由于利用了顶点内并行性，因此单个顶点的顶点延迟要小于同时处理多个顶点。

（1.4）添加一个缓冲区。在聚合阶段添加缓冲区，预先将一些顶点度数高于阈值的顶点预取到缓冲区中，阈值设定由图大小所决定。在进行预取时也能够进行图划分（静态），从而提高并行率，且图数据访问时大多数情况下均访问高度数顶点，因此利用了图数据访问的空间局部性，提高访存命中率。

（1.5）在聚合阶段完成处理后将处理结果传入流水线，以便组合引擎读取数据。

（2）组合引擎。采用收缩阵列来处理聚合引擎中传入的数据，并通过使用颗粒的大小将其分为独立工作模式和合作工作模式。其主要包括如下子步骤：

（2.1）组合引擎主要是由收缩阵列组成，为了适应聚合引擎的两种处理模式，组合引擎集成了多个阵列，允许多颗粒地使用这些收缩模块，因此组合引擎分为独立的工作模式和合作的工作模式。

（2.2）独立的工作模式。在这种模式下，收缩期模块相互独立地工作。每个模块都处理一小组顶点的矩阵乘法操作，每个模块的权重参数都可以直接从权重缓冲区中访问，并仅在模块中重用。这种模式的优点是顶点延迟较低，因为我们可以在这一小群顶点的聚合特性准备好后立即处理它们的组合操作，而无需等待更多的顶点。

（2.3）合作的工作模式。这些收缩期模块还可以进一步组装在一起，同时处理更多的顶点。这种模式需要在执行组合操作之前将大量顶点的聚合特征组合在一起。其优点是，重量参数可以从重量缓冲器流到下游收缩模块，然后逐渐流到上游模块，所有收缩阵列都可以重复使用。

（2.4）在每次组合引擎完成计算后判断该次迭代是否为最后一次迭。若为最后一次迭代则将结果进行输出，结束图卷积神经网络；若部位最后一次迭代，将中间结果传回缓冲区，由聚合阶段重新提取，回到（1.2）。

（3）聚合组合缓冲区。为了重用聚合引擎产生的聚合结果，在两个引擎间添加了聚合组合缓冲区，该缓冲区可由聚合引擎写入，也可由组合引擎读取。其主要包括如下子步骤：

（3.1）延迟感知流水线。在此流水线模式下，组合引擎在收缩期模块独立模式下工作。聚合特性在聚合引擎中逐顶点生成，一旦一小群顶点的聚合特性准备好，将立即处理以下组合。因此，每个顶点的平均处理延迟可以更低。

（3.2）能量感知流水线。能量感知流水线在组合发动机中使用收缩期模块协同模式。逐顶点的处理将更改为突发模式，即每次都会一起处理大量的顶点。虽然顶点延迟较长，但由于在合并的收缩期阵列中没有冗余访问的权重传播，可以减少能量消耗。

（4）芯片外内存的访问协调。由于聚合引擎和组合引擎的实际工作负载通常不同，因此为其各分配一个芯片外内存的方法会造成带宽浪费。而两个引擎在运行时同时访问内存，导致访问位置的频繁切换，因此预定义了一个优先级来进行访问。其主要包括如下子步骤：

（4.1）预先定义了一个访问优先级（边缘>输入特征>权重> 输出特征）来组装不连续请求，访问请求被逐批执行。因此，当前批中的低优先级访问是在下一批中的高优先级访问之前处理的，而不是总是首先进行高优先级访问。随着连续性的提高，行缓冲区的利用率可以显著提高。接下来，我们重新映射这些重新排序的地址，以使用低位来索引通道和块。从而进一步利用内存通道级和块级的并行性。

本专利相比现有的图卷积神经网络架构均有显著提高。在PyTorch的几何图形中，本专利与PyGCPU和PyG-GPU在加速，能耗，动态随机存取内存带宽利用率和动态随机存取内存访问方面均有较大提升。动态随机存取内存访问方面的提升主要原因在于本专利进行的稀疏性消除，减少了冗余访问。动态随机存取内存带宽利用率的提升则主要受益于芯片外内存访问的协调，更好地利用了通道/块级地并行性。

本领域的技术人员容易理解，以上所述仅为本发明的较佳实施对比而已，并不用以限制本发明，凡在本发明的精神和原则之内所作任何修改，等同替换和改进等，均应包含在本发明的保护范围之内。

**说 明 书 附 图**

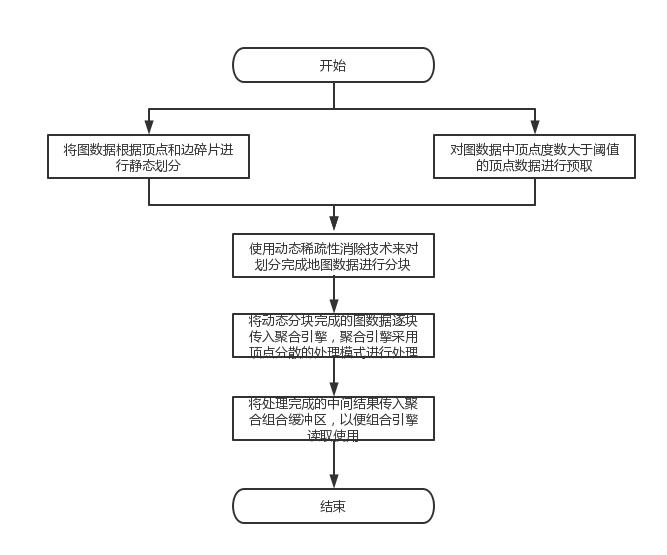


图1

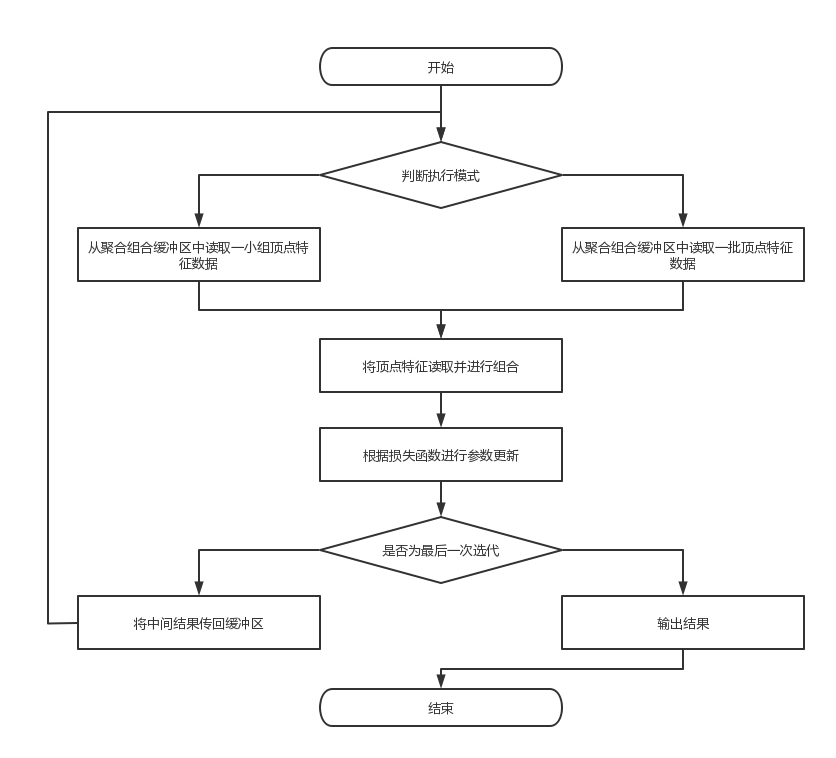


图2

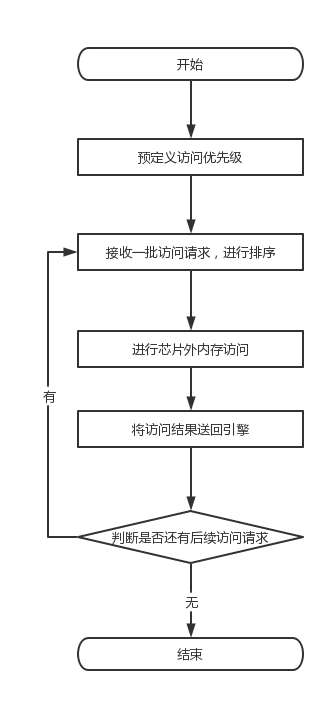


图3