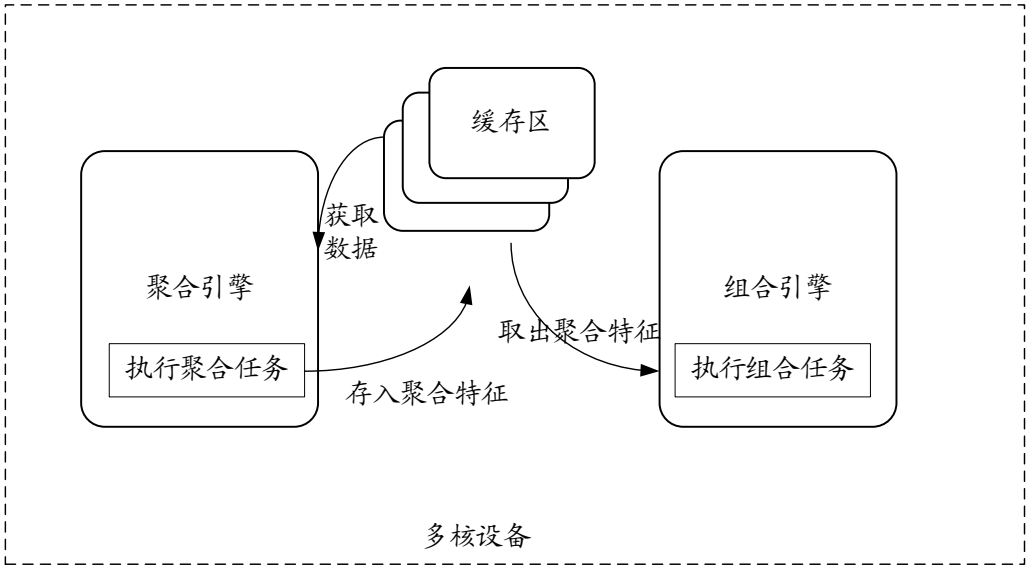


## 说明书摘要

本说明书公开了一种针对图卷积加速的执行系统，系统包括：聚合引擎、组合引擎、缓存区，系统可以通过多核设备运行聚合引擎和组合引擎，聚合引擎与组合引擎并行处理任务，聚合引擎获取原始图数据，将原始图数据进行划分，得到若干子图数据，并从若干子图数据中，选取出至少部分子图数据存储  
5 在缓存区中，根据缓存区中的子图数据，构建各聚合任务，并将每个聚合任务分配给各核，不同核之间并行处理各聚合任务，在聚合任务完成后将得到的节点对应的聚合特征存入缓存区中；组合引擎可以针对每个节点，从缓存区中获取该节点的聚合特征，并根据该节点的历史节点特征以及对应的聚合特征，执行  
10 组合任务，从而，本系统可以在一定程度上提高图卷积的效率。

摘要附图



5

## 权 利 要 求 书

1、一种针对图卷积加速的执行系统，其特征在于，所述系统包括：聚合引擎、组合引擎、缓存区，其中，所述系统通过多核设备运行所述聚合引擎和所述组合引擎，所述聚合引擎与所述组合引擎并行处理任务；

5        所述聚合引擎用于，获取原始图数据，将所述原始图数据进行划分，得到若干子图数据，并从所述若干子图数据中，选取出至少部分子图数据存储在缓存区中，根据所述缓存区中的子图数据，构建针对所述原始图数据中包含的节点的各聚合任务，并将得到的每个聚合任务分配给各核，不同核之间并行处理各聚合任务，在聚合任务完成后将得到的节点对应的聚合特征存入所述缓存区  
10    中，其中，针对每个节点，该节点对应的聚合特征是通过对该节点具有预设邻接关系的节点的节点特征聚合得到的；

所述组合引擎用于，针对每个节点，从所述缓存区中获取该节点的聚合特征，并根据该节点的历史节点特征以及对应的聚合特征，执行组合任务。

2、如权利要求 1 所述的系统，其特征在于，所述组合引擎用于，执行组合任务得到此次迭代中该节点的节点特征后，确定此次迭代是否为最后一次迭代，  
15    若否，则将该节点特征存入到所述缓存区中，以使所述聚合引擎继续针对该节点执行聚合任务。

3、如权利要求 1 所述的系统，其特征在于，一个子图数据包括：节点区间以及边区间，节点区间用于表示该子图数据对应的子图中包含的节点，所述  
20    边区间用于表示该子图数据对应子图中包含的节点对应的边，所述节点区间和所述边区间分别通过压缩稀疏列进行表示。

4、如权利要求 1 所述的系统，其特征在于，所述系统中还包括：内存，所述内存用于存储所述若干子图数据；

所述聚合引擎用于，确定每个节点的度，从所述若干子图数据中，选取出  
25    度的数值高于预设数值的节点对应的子图数据存储在所述缓存区中；根据所述缓存区中的子图数据，以及所述内存中的子图数据，构建所述各聚合任务；

若需要针对节点执行聚合任务，确定所述缓存区中是否存在与该节点相关的子图数据，若是，根据所述子图数据构建该节点对应的聚合任务，若否，从所述内存中读取与该节点相关的子图数据，以确定出与该节点相关的聚合元素，并根据与该节点相关的聚合元素，构建出该节点的聚合任务。

5 5、如权利要求 3 所述的系统，其特征在于，所述聚合引擎用于，针对每个边区间，在该边区间内创建预设大小的窗口，所述窗口向下滑动，直到所述窗口顶部一行内存在有边，所述窗口停止，并且所述窗口的底部一行向上收缩，直到所述窗口的底部一行内存在有边。将该边区间按照所述窗口的顶部一行以及所述窗口的底部一行进行调整，得到调整后的边区间，并将调整后的边区间进行存储。

删除的内容：第一

删除的内容：第一

删除的内容：该第一

删除的内容：第一

删除的内容：则在第一窗口下方再创建所述预设大小的第二窗口，第二窗口向下滑动，直到该第二窗口的内存存在有边，第二窗口停止，并且第二窗口的

删除的内容：第二

删除的内容：第一

删除的内容：第二窗口的

删除的内容：，

10 6、如权利要求 4 所述的系统，其特征在于，所述缓存区中包括用于存储度的数值高于预设数值的节点对应的子图数据的第一缓存区，还包括用于存储聚合特征的第二缓存区。

15 7、如权利要求 1 所述的系统，其特征在于，所述组合引擎中包含有若干收缩阵列，所述组合引擎包括独立工作模式以及合作工作模式；

在所述独立工作模式下，不同节点的组合任务在所述组合引擎中的不同收缩阵列中执行；

在所述合作工作模式下，所述组合引擎中的各收缩阵列共同执行节点的组合任务。

20 8、如权利要求 7 所述的系统，其特征在于，若所述组合引擎为所述独立工作模式，所述聚合引擎执行完成少于第一设定数量的聚合任务后，将得到的少于所述第一设定数量的聚合特征存入所述缓存区；

所述组合引擎监测到所述缓存区中存储有少于第一设定数量的聚合特征，获取到所述少于第一设定数量的聚合特征，按照所述独立工作模式通过收缩阵列执行组合任务。

25 9、如权利要求 7 所述的系统，其特征在于，若所述组合引擎为所述组合

工作模式，所述聚合引擎执行完成不少于第二设定数量的聚合任务后，将得到的少于所述第二设定数量的聚合特征存入所述缓存区；

所述组合引擎监测到所述缓存区中存储有不少于第二设定数量的聚合特征，获取到所述不少于第二设定数量的聚合特征，按照所述合作工作模式通过  
5 收缩阵列执行组合任务。

10、如权利要求 4 所述的系统，其特征在于，所述系统针对所述内存以及所述缓存区预设访问优先级。

11、一种针对图卷积加速的执行方法，其特征在于，所述执行方法通过包括有聚合引擎、组合引擎、缓存区以及内存的系统进行执行，所述系统通过多  
10 核设备运行所述聚合引擎和所述组合引擎，所述聚合引擎与所述组合引擎并行处理任务，包括：

获取原始图数据；

通过所述聚合引擎将所述原始图数据进行划分，得到若干子图数据；

从所述若干子图数据中，选取至少部分子图数据存储在缓存区中；

15 通过所述聚合引擎根据所述缓存区中的子图数据，构建针对所述原始图数据中包含的节点的各聚合任务，并将得到的每个聚合任务分配给各核，不同核之间并行处理各聚合任务；

在聚合任务完成后将得到的节点对应的聚合特征存入所述缓存区中，其中，针对每个节点，该节点对应的聚合特征是通过对该节点具有预设邻接关系的节点的节点特征聚合得到的；  
20

通过所述组合引擎针对每个节点，从所述缓存区中获取该节点的聚合特征，并根据该节点的历史节点特征以及对应的聚合特征，执行组合任务。

12、一种针对图卷积加速的执行装置，其特征在于，所述执行装置中包括聚合引擎、组合引擎、缓存区，所述装置通过多核设备运行所述聚合引擎和所  
25 述组合引擎，所述聚合引擎与所述组合引擎并行处理任务，包括：

获取模块，用于获取原始图数据；

划分模块，用于通过所述聚合引擎将所述原始图数据进行划分，得到若干子图数据；

选取模块，用于从所述若干子图数据中，选取出至少部分子图数据存储在缓存区中；

- 5       构建模块，用于通过所述聚合引擎根据所述缓存区中的子图数据，构建针对所述原始图数据中包含的节点的各聚合任务，并将得到的每个聚合任务分配给各核，不同核之间并行处理各聚合任务；

- 10       存储模块，用于在聚合任务完成后将得到的节点对应的聚合特征存入所述缓存区中，其中，针对每个节点，该节点对应的聚合特征是通过对该节点具有预设邻接关系的节点的节点特征聚合得到的；

执行模块，用于通过所述组合引擎针对每个节点，从所述缓存区中获取该节点的聚合特征，并根据该节点的历史节点特征以及对应的聚合特征，执行组合任务。

- 15       13、一种计算机可读存储介质，其特征在于，所述存储介质存储有计算机程序，所述计算机程序被处理器执行时实现上述权利要求 1 所述的方法。

14、一种电子设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，其特征在于，所述处理器执行所述程序时实现上述权利要求 1 所述的方法。

# 说明书

## 一种针对图卷积加速的执行系统

### 技术领域

本说明书涉及神经网络技术领域，尤其涉及一种针对图卷积加速的执行系统。

### 背景技术

当前，图（Graph）数据可以应用在多种领域中，如信息推荐、链路预测等，为有效地挖掘图数据的特性，业界提出了图卷积神经网络。图卷积神经网络将图数据转化到一个低维空间，同时能够最大化保持结构和属性信息，为后续通过图数据的特征来执行相关业务，提供了有效的途径。

图卷积有两个主要的执行阶段：聚合和组合。聚合阶段依赖于固有的稀疏的图结构。针对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。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。在常规的图卷积处理中，通常将整个图存在磁盘或者内存中，在需要进行节点聚合时，需要加载出整个图，并从整个图中搜索出需要的节点等相关数据，从而开始进行聚合、组合等，但是这种方式需要加载出整个图，效率较低。

因此，如何提高进行图卷积的效率，则是一个亟待解决的问题。

### 发明内容

本说明书提供一种针对图卷积加速的执行系统，以部分的解决现有技术存在的上述问题。

本说明书采用下述技术方案：

本说明书提供了一种针对图卷积加速的执行系统，所述系统包括：聚合引擎、组合引擎、缓存区，其中，所述系统通过多核设备运行所述聚合引擎和所

述组合引擎，所述聚合引擎与所述组合引擎并行处理任务；

所述聚合引擎用于，获取原始图数据，将所述原始图数据进行划分，得到若干子图数据，并从所述若干子图数据中，选取出至少部分子图数据存储在缓存区中，根据所述缓存区中的子图数据，构建针对所述原始图数据中包含的节点  
5 的各聚合任务，并将得到的每个聚合任务分配给各核，不同核之间并行处理各聚合任务，在聚合任务完成后将得到的节点对应的聚合特征存入所述缓存区中，其中，针对每个节点，该节点对应的聚合特征是通过对该节点具有预设邻接关系的节点的节点特征聚合得到的；

所述组合引擎用于，针对每个节点，从所述缓存区中获取该节点的聚合特征，并根据该节点的历史节点特征以及对应的聚合特征，执行组合任务。  
10

可选地，所述组合引擎用于，执行组合任务得到此次迭代中该节点的节点特征后，确定此次迭代是否为最后一次迭代，若否，则将该节点特征存入到所述缓存区中，以使所述聚合引擎继续针对该节点执行聚合任务。

可选地，一个子图数据包括：节点区间以及边区间，节点区间用于表示该子图数据对应的子图中包含的节点，所述边区间用于表示该子图数据对应子图中包含的节点对应的边，所述节点区间和所述边区间分别通过压缩稀疏列进行表示。  
15

可选地，所述系统中还包括：内存，所述内存用于存储所述若干子图数据；

所述聚合引擎用于，确定每个节点的度，从所述若干子图数据中，选取出度的数值高于预设数值的节点对应的子图数据存储在所述缓存区中；根据所述缓存区中的子图数据，以及所述内存中的子图数据，构建所述各聚合任务；  
20

若需要针对节点执行聚合任务，确定所述缓存区中是否存在与该节点相关的子图数据，若是，根据所述子图数据构建该节点对应的聚合任务，若否，从所述内存中读取与该节点相关的子图数据，以确定出与该节点相关的聚合元素，并根据与该节点相关的聚合元素，构建出该节点的聚合任务。  
25

可选地，所述聚合引擎用于，针对每个边区间，在该边区间内创建预设大



小的窗口，所述窗口向下滑动，直到所述窗口顶部一行内存在有边，所述窗口停止，并且所述窗口的底部一行向上收缩，直到所述窗口的底部一行内存在有边。将该边区间按照所述窗口的顶部一行以及所述窗口的底部一行进行调整，得到调整后的边区间，并将调整后的边区间进行存储。

5        可选地，所述缓存区中包括用于存储度的数值高于预设数值的节点对应的子图数据的第一缓存区，还包括用于存储聚合特征的第二缓存区。

可选地，所述组合引擎中包含有若干收缩阵列，所述组合引擎包括独立工作模式以及合作工作模式；

10        在所述独立工作模式下，不同节点的组合任务在所述组合引擎中的不同收缩阵列中执行；

在所述合作工作模式下，所述组合引擎中的各收缩阵列共同执行节点的组合任务。

15        可选地，若所述组合引擎为所述独立工作模式，所述聚合引擎执行完成少于第一设定数量的聚合任务后，将得到的少于所述第一设定数量的聚合特征存入所述缓存区；

所述组合引擎监测到所述缓存区中存储有少于第一设定数量的聚合特征，获取到所述少于第一设定数量的聚合特征，按照所述独立工作模式通过收缩阵列执行组合任务。

20        可选地，若所述组合引擎为所述组合工作模式，所述聚合引擎执行完成不少于第二设定数量的聚合任务后，将得到的少于所述第二设定数量的聚合特征存入所述缓存区；

所述组合引擎监测到所述缓存区中存储有不少于第二设定数量的聚合特征，获取到所述不少于第二设定数量的聚合特征，按照所述合作工作模式通过收缩阵列执行组合任务。

25        可选地，所述系统针对所述内存以及缓存区预设有访问优先级。

本说明书中提供一种针对图卷积加速的执行方法，所述执行方法通过包括

有聚合引擎、组合引擎、缓存区以及内存的系统进行执行，所述系统通过多核设备运行所述聚合引擎和所述组合引擎，所述聚合引擎与所述组合引擎并行处理任务，包括：

获取原始图数据；

5 通过所述聚合引擎将所述原始图数据进行划分，得到若干子图数据；

从所述若干子图数据中，选取出至少部分子图数据存储在缓存区中；

通过所述聚合引擎根据所述缓存区中的子图数据，构建针对所述原始图数据中包含的节点的各聚合任务，并将得到的每个聚合任务分配给各核，不同核之间并行处理各聚合任务；

10 在聚合任务完成后将得到的节点对应的聚合特征存入所述缓存区中，其中，针对每个节点，该节点对应的聚合特征是通过对该节点具有预设邻接关系的节点的节点特征聚合得到的；

通过所述组合引擎针对每个节点，从所述缓存区中获取该节点的聚合特征，并根据该节点的历史节点特征以及对应的聚合特征，执行组合任务。

15 本说明书提供一种针对图卷积加速的执行装置，所述执行装置中包括聚合引擎、组合引擎、缓存区，所述装置通过多核设备运行所述聚合引擎和所述组合引擎，所述聚合引擎与所述组合引擎并行处理任务，包括：

获取模块，用于获取原始图数据；

20 划分模块，用于通过所述聚合引擎将所述原始图数据进行划分，得到若干子图数据；

选取模块，用于从所述若干子图数据中，选取出至少部分子图数据存储在缓存区中；

25 构建模块，用于通过所述聚合引擎根据所述缓存区中的子图数据，构建针对所述原始图数据中包含的节点的各聚合任务，并将得到的每个聚合任务分配给各核，不同核之间并行处理各聚合任务；

存储模块，用于在聚合任务完成后将得到的节点对应的聚合特征存入所述

缓存区中，其中，针对每个节点，该节点对应的聚合特征是通过对该节点具有预设邻接关系的节点的节点特征聚合得到的；

执行模块，用于通过所述组合引擎针对每个节点，从所述缓存区中获取该节点的聚合特征，并根据该节点的历史节点特征以及对应的聚合特征，执行组合任务。

本说明书提供了一种计算机可读存储介质，所述存储介质存储有计算机程序，所述计算机程序被处理器执行时实现上述针对图卷积加速的执行方法。

本说明书提供了一种电子设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，所述处理器执行所述程序时实现上述针对图卷积加速的执行方法。

本说明书采用的上述至少一个技术方案能够达到以下有益效果：

从上述针对图卷积加速的执行系统中可以看出，系统包括：聚合引擎、组合引擎、缓存区，其中，系统可以通过多核设备运行聚合引擎和组合引擎，聚合引擎与组合引擎并行处理任务，聚合引擎用于，获取原始图数据，将原始图数据进行划分，得到若干子图数据，并从若干子图数据中，选取出至少部分子图数据存储于缓存区中，根据缓存区中的子图数据，构建针对原始图数据中包含的节点的各聚合任务，并将得到的每个聚合任务分配给各核，不同核之间并行处理各聚合任务，在聚合任务完成后将得到的节点对应的聚合特征存入缓存区中，其中，针对每个节点，该节点对应的聚合特征是通过对该节点具有预设邻接关系的节点的节点特征聚合得到的；组合引擎用于，针对每个节点，从所述缓存区中获取该节点的聚合特征，并根据该节点的历史节点特征以及对应的聚合特征，执行组合任务。

从上述内容中可以看出，本说明书提供的系统可以，将原始图数据进行划分，从而得到多个子图数据，缓存区中可以存储一部分子图数据，在聚合引擎与组合引擎执行任务时，可以先从缓存区中读取所需的数据，从而提高缓存命中率，并且，聚合引擎和组合引擎可以并行执行任务，在聚合引擎执行聚合任

务的同时，组合引擎读取缓存区中所存储的聚合特征并可以开始执行组合任务，从而，在一定程度上提高了图卷积的效率。

### 附图说明

此处所说明的附图用来提供对本说明书的进一步理解，构成本说明书的一部分，本说明书的示意性实施例及其说明用于解释本说明书，并不构成对本说明书的不当限定。在附图中：

图 1 为本说明书提供了一种针对图卷积加速的执行系统的示意图；

图 2 为本说明书中提供了一种聚合引擎执行过程的流程示意图；

图 3 为本说明书中提供了一种组合引擎执行的流程示意图；

图 4 为本说明书提供了一种针对图卷积加速的执行装置示意图；

图 5 为本说明书提供的对应于图 1 的电子设备示意图。

### 具体实施方式

为使本说明书的目的、技术方案和优点更加清楚，下面将结合本说明书具体实施例及相应的附图对本说明书技术方案进行清楚、完整地描述。显然，所描述的实施例仅是本说明书一部分实施例，而不是全部的实施例。基于本说明书中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本说明书保护的范围。

以下结合附图，详细说明本说明书各实施例提供的技术方案。

图 1 为本说明书提供了一种针对图卷积加速的执行系统的示意图，该系统中可以包括：聚合引擎、组合引擎、缓存区，以及内存等，其中，该系统可以通过多核设备运行所述聚合引擎和所述组合引擎，所述聚合引擎与所述组合引擎并行处理任务。

在图卷积中，针对图数据的节点特征提取操作，是迭代进行的，图数据中每个节点可以存在初始的特征，在一轮迭代中，随着聚合与组合的进行，节点将被更新一次特征，即，存在多少轮迭代，节点的特征将被更新多少次，直到停止迭代。

本说明书中的聚合引擎与组合引擎可以并行处理任务，顾名思义，聚合引擎主要处理的是与聚合相关的任务，组合引擎主要处理的是与组合相关的任务，两个引擎之间可以存在一个缓存区（在下面的内容中将该缓存区称之为：第二缓存区）。当聚合引擎完成一个或多个聚合任务，得到针对一个或多个节点5 的聚合特征后，可以将聚合特征传入该缓存区中。组合引擎可以通过缓存区中的聚合特征，开始进行针对节点的组合操作，即，通过聚合特征，对节点在上一次迭代中得到的节点特征（若此次是第一次迭代则是节点的初始特征）进行更新。

由于聚合引擎与组合引擎分开，聚合操作和组合操作之间解耦，使得组合引擎在执行组合任务的过程中，聚合引擎依然可以执行聚合任务，两者之间并不是完全串行的，即，在一小部分节点的聚合任务完成后，组合引擎可以开始执行这一小部分节点的组合任务，与此同时，聚合引擎可以开始执行另一部分节点的聚合任务，以此类推，整个图卷积过程，聚合引擎和组合引擎之间处理任务均可以是并行的，从而能够在一定程度上提高图卷积的效率。

15 在本系统中，为了进一步地提高效率，在首次迭代前，聚合引擎可以获取原始图数据，将所述原始图数据进行划分，得到若干子图数据。

而后，聚合引擎可以从若干子图数据中，选取出至少部分子图数据存储在缓存区中，根据缓存区中的子图数据，构建针对原始图数据中包含的节点的各项聚合任务，并将得到的每个聚合任务分配给各核，不同核之间并行处理各聚合任务，在聚合任务完成后将得到的节点对应的聚合特征存入缓存区中，其中，20 针对每个节点，该节点对应的聚合特征是通过对该节点具有预设邻接关系的节点的节点特征聚合得到的。

组合引擎可以，针对每个节点，从缓存区中获取该节点的聚合特征，并根据该节点的历史节点特征以及对应的聚合特征，执行组合任务。在执行组合任务得到此次迭代中该节点的节点特征后，可以确定此次迭代是否为最后一次25 迭代，若否，则可以将该节点特征存入到缓存区中，以使聚合引擎继续针对该

节点执行聚合任务。

上述提到的子图数据中可以包括节点区间以及边区间，节点区间和边区间可以分别通过压缩稀疏列进行表示，其中，一个节点区间可以表示该子图数据对应的子图中包含的节点，边区间可以用于表示该子图数据对应子图中包含的节点对应的边。

为了进一步地压缩存储子图数据所需的空间，可以针对每个边区间，将该边区间压缩，得到压缩后的边区间进行存储。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来压缩边区间的方式，目的是将边区间中不需要存储的部分数据去掉。

针对每个边区间，可以在该边区间内创建预设大小的窗口，该窗口向下滑动，直到窗口顶部一行内存在有边，该窗口停止，并且该窗口的底部一行向上收缩，直到该窗口的底部一行内存在有边。将该边区间按照该窗口的顶部一行以及该窗口的底部一行进行调整，得到调整后的边区间，并将调整后的边区间进行存储。

将该边区间按照窗口的顶部一行以及底部一行进行调整，得到调整后的边区间，也就是说，调整后的边区间的外围中不存在未含有任何边的一行。

其中，窗口为矩阵形式，边区间为邻接矩阵形式，在存储调整后的边区间时可以通过稀疏压缩列的形式进行存储，而在调整该边区间时，该边区间可以是邻接矩阵的形式。在执行聚合任务或者组合任务时，可以直接使用调整后的边区间得到所需的数据。

在调整上述边区间时，可以是针对原始的邻接矩阵中每个节点区间对应的边区间进行调整，而窗口大小可以与原始的边区间大小相同。

下面以一个例子的形式，来对节点区间以及边区间进行举例。例如，共有[1、2、3、4]四个节点，四个节点之间的边可以由 $4 \times 4$ 的邻接矩阵表示，节点区间可以包括[1、2]两个节点，[3、4]两个节点，边区间可以是上述 $4 \times 4$ 的邻接矩阵被分为的一些小矩阵，例如，可以分为4个 $2 \times 2$ 的小矩阵，一个小矩

删除的内容: 可以创建在该边区间内预设大小的第一窗口，第一窗口向下滑动，直到该第一窗口顶部一行内存在有边，第一窗口停止，则在第一窗口下方再创建所述预设大小的第二窗口，第二窗口向下滑动，直到该第二窗口的内存在有边，第二窗口停止，并且第二窗口的底部一行向上收缩，直到第二窗口的底部一行内存在有边

删除的内容: 第一

删除的内容: 第二窗口的

删除的内容: ，并将调整后的边区间进行存储

删除的内容: 第一

删除的内容: 与第二窗口

阵为两个节点与两个节点之间边的情况，再例如，可以分为 2 个  $4 \times 2$  的小矩阵，每个小矩阵包含一种节点区间中节点的所有边。

而若分出的边区间对应的小矩阵中包含有多余的一行（即，这一行中不包含边），则可以通过上述基于窗口的滑动和收缩来压缩边区间的方式，将边区间中多余的这种行去除。

也就是说，在本说明书中，可以将原始的图数据，划分为多个子图数据，多个子图数据可以被存储在内存中或磁盘中，缓存区中可以存储部分子图数据，在执行聚合任务时，聚合引擎可以先从缓存区中获取到需要的子图数据，从而确定出用于节点聚合所需的数据。并且，由于图是稀疏的，拆分为子图数据后，每个子图数据中可能会存在有不需要存储的数据，还可以针对子图数据中的边区间进行调整，从而降低需要存储的数据量。

需要说明的是，缓存区中存储的部分子图数据可以是指度的数值高于预设数值的节点对应的子图数据，从而，内存中可以存储全部子图数据（当然也可以存储除上述至少部分子图数据外的剩余子图数据），在需要执行聚合任务时，可以先从缓存区中确定节点聚合所需的数据，若缓存区中不存在，则可以再到内存中确定节点聚合所需的数据，如图 2 所示。

图 2 为本说明书中提供一种聚合引擎执行过程的流程示意图。

从图 2 中可以看出，聚合引擎可以将原始图数据进行拆分，得到多个子图数据，并且，可以基于上述窗口的滑动和收缩的方式，来将子图数据中的边区间进行调整，从而减少存储的边区间所需的空间。

缓存区中还可以存在有专门的缓存区，该缓存区用于存储度的数值高于设定数值的节点对应的子图数据，其中，这里提到的度的数值高于设定数值的节点对应的子图数据，可以是指度的数值高于设定数值的节点在子图数据中所对应的那一小部分边的数据，以及该缓存区还可以存储与该节点相关的数据，如特征数据等。

在将原始图数据划分好，并且缓存区也准备好相应的数据后，聚合引擎可

以开始执行聚合任务，并在聚合任务完成后，将得到的聚合特征存储到缓存区中。

其中，可以将一个子图数据存储到缓存区（专门用于缓存子图数据的第三缓存区）中，使得聚合引擎可以先对该子图数据中的节点执行聚合任务，聚合引擎将该子图数据中的节点执行聚合任务完成后，再将缓存区中的子图数据替换。

若需要针对节点执行聚合任务，可以确定缓存区中是否存在与该节点相关的子图数据，若是，根据子图数据构建该节点对应的聚合任务，若否，从内存中读取与该节点相关的子图数据，以确定出与该节点相关的聚合元素，并根据与该节点相关的聚合元素，构建出该节点的聚合任务。这里提到的聚合元素可以是指与节点的聚合任务所需的数据。

其中，上述度可以是指与节点相连的边的数量。可以专门存在一个缓存区：在此称之为第一缓存区，用于存储度的数值高于预设数值的节点对应的子图数据，专门存在该第一缓存区的作用为，在执行聚合任务时，若该第一缓存区中存在所需的数据，则不需要再通过内存进行获取，从而能够提高访存命中率，进而提高图卷积的效率。

还需说明的是，在聚合引擎或者组合引擎需要从内存或缓存中查找数据时，可以先判断所需查找的数据是否是度的数值高于预设数值的节点的相关数据，若是，则可以直接从缓存区中查找所需的数据，若否，则可以从内存中查找所需的数据。

需要说明的是，从内存中查找所需的数据，可以从内存中取出所需的数据所位于的边区间，进而，将边区间存储在第三缓存区中，再通过第三缓存区中的边区间，查找出所需的数据。

在本说明书中，组合引擎中包含有若干收缩阵列，该组合引擎可以对应有两种工作模式：独立工作模式以及合作工作模式，当开发人员需要更加有组合引擎效率地完成组合任务，则可以选择独立工作模式，当开发人员需要组合引



引擎完成组合任务的能耗更低，则可以选择合作工作模式。

独立工作模式下，不同节点的组合任务在组合引擎中的不同收缩阵列中执行，即，一个收缩阵列可以同于执行一个组合任务。在合作工作模式下，组合引擎中的各收缩阵列可以共同执行节点的组合任务，即，各收缩阵列可以合作完成一个或多个组合任务。

针对两种工作模式，缓存区接收聚合引擎的方式，或者说聚合引擎将聚合特征存储到缓存区的方式可以有所不同。

若组合引擎为独立工作模式，聚合引擎执行完成少于第一设定数量的聚合任务后，则可以将得到的少于第一设定数量的聚合特征存入所述缓存区，组合引擎监测到缓存区中存储有少于第一设定数量的聚合特征，获取到少于第一设定数量的聚合特征，按照独立工作模式通过收缩阵列执行组合任务。

若组合引擎为组合工作模式，聚合引擎执行完成不少于第二设定数量的聚合任务后，则可以将得到的少于第二设定数量的聚合特征存入缓存区，组合引擎监测到缓存区中存储有不少于第二设定数量的聚合特征，获取到不少于第二设定数量的聚合特征，按照合作工作模式通过收缩阵列执行组合任务。第一设定数量可以少于第二设定数量。第一设定数量与第二设定数量均可以被人为设定。

也就是说，在独立工作模式下，得到了少量的聚合特征后，即可以存储在缓存区中，便于组合引擎中的各收缩阵列并行执行组合任务，而合作模式下，则可以在得到较多的聚合特征后，再存储在缓存区中，使得组合引擎中收缩阵列一次性合作地执行多个组合任务

下面以一种例子的形式，来说明组合引擎的工作流程，如图 3 所示。

图 3 为本说明书中提供的一种组合引擎执行的流程示意图。

从图 3 中可以看出，首先组合引擎可以判断自身是独立工作模式还是组合工作模式，若是独立工作模式，则可以少量少量地从缓存区中读取出聚合特征，开始执行组合任务，而若是组合工作模式，则可以从缓存区中读取出较多的聚

合特征，开始执行组合任务。

执行组合任务后，可以根据损失函数对图卷积的模型进行参数更新，并判断当前是否为最后一次迭代，若不是，则需要将组合得到的节点特征传回缓存区，若是最后一次迭代，则可以输出组合得到的节点特征，并在输出最后一个节点的节点特征后，结束任务。

需要说明的是，在设置上述边区间的大小时，该边区间的大小可以与缓存区的大小相关，边区间高度由第一缓存区的容量决定，而边区间宽度由第二缓冲区的容量决定。第三缓存区的大小可以影响边区间的高度和宽度，因为它可以容纳整个边区间的所包含的每个边。

在本说明书中，聚合引擎以及组合引擎可以共用内存以及缓存区，针对内存以及缓存区可以预设有访问优先级，具体该访问优先级可以为：边>聚合特征>权重>节点特征。例如，在一批访问请求中，可以先处理高优先级的访问请求，再处理低优先级的访问请求，当前一批中的低优先级访问可以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。一批访问请求可以是连续一定时间内访问内存或者缓存的请求，也可以是连续一定数量的访问内存或者缓存的请求。

从上述内容中可以看出，本说明书提供的针对图卷积加速的执行系统可以，提高图卷积的效率，由于将原始图数据进行拆分，拆分后的子图数据中的边区间可以按照上述窗口的滑动及收缩的方式进行压缩，从而去除边区间中的多余数据，在一定程度上消除图的稀疏性。

并且，可以将高度数的节点（度的数值高于预设数值的节点）的相关数据存储在独立的缓存区中，以及逐个将拆分后的子图数据存储在缓存区中，能够在一定程度上提高访问命中的速度。

图4为本说明书提供的一种针对图卷积加速的执行装置示意图，所述执行装置中包括聚合引擎、组合引擎、缓存区，所述执行装置通过多核设备运行所述聚合引擎和所述组合引擎，所述聚合引擎与所述组合引擎并行处理任务，包

括:

获取模块 401, 用于获取原始图数据;

划分模块 402, 用于通过所述聚合引擎将所述原始图数据进行划分, 得到若干子图数据;

5 选取模块 403, 用于从所述若干子图数据中, 选取至少部分子图数据存储在缓存区中;

构建模块 404, 用于通过所述聚合引擎根据所述缓存区中的子图数据, 构建针对所述原始图数据中包含的节点的各聚合任务, 并将得到的每个聚合任务分配给各核, 不同核之间并行处理各聚合任务;

10 存储模块 405, 用于在聚合任务完成后将得到的节点对应的聚合特征存入所述缓存区中, 其中, 针对每个节点, 该节点对应的聚合特征是通过对该节点具有预设邻接关系的节点的节点特征聚合得到的;

执行模块 406, 用于通过所述组合引擎针对每个节点, 从所述缓存区中获取该节点的聚合特征, 并根据该节点的历史节点特征以及对应的聚合特征, 执行  
15 行组合任务。

可选地, 所述执行模块 406 用于, 执行组合任务得到此次迭代中该节点的节点特征后, 确定此次迭代是否为最后一次迭代, 若否, 则将该节点特征存入到所述缓存区中, 以使所述聚合引擎继续针对该节点执行聚合任务。

可选地, 一个子图数据包括: 节点区间以及边区间, 节点区间用于表示该子图数据对应的子图中包含的节点, 所述边区间用于表示该子图数据对应子图中包含的节点对应的边, 所述节点区间和所述边区间分别通过压缩稀疏列进行表示。  
20

可选地, 所述执行装置中还包括: 内存, 所述内存用于存储所述若干子图数据;

25 所述构建模块 404 用于, 确定每个节点的度, 从所述若干子图数据中, 选取度的数值高于预设数值的节点对应的子图数据存储在所述缓存区中; 根据

所述缓存区中的子图数据,以及所述内存中的子图数据,构建所述各聚合任务;

若需要针对节点执行聚合任务,确定所述缓存区中是否存在与该节点相关的子图数据,若是,根据所述子图数据构建该节点对应的聚合任务,若否,从所述内存中读取与该节点相关的子图数据,以确定出与该节点相关的聚合元素,并根据与该节点相关的聚合元素,构建出该节点的聚合任务。

可选地,所述划分模块 402 还用于,针对每个边区间, 在该边区间内创建预设大小的窗口,所述窗口向下滑动,直到所述窗口顶部一行内存在有边,所述窗口停止,并且所述窗口的底部一行向上收缩,直到所述窗口的底部一行内存在有边。将该边区间按照所述窗口的顶部一行以及所述窗口的底部一行进行调整,得到调整后的边区间,并将调整后的边区间进行存储。,

可选地,所述缓存区中包括用于存储度的数值高于预设数值的节点对应的子图数据的第一缓存区,还包括用于存储聚合特征的第二缓存区。

可选地,所述组合引擎中包含有若干收缩阵列,所述组合引擎包括独立工作模式以及合作工作模式;在所述独立工作模式下,不同节点的组合任务在所述组合引擎中的不同收缩阵列中执行;在所述合作工作模式下,所述组合引擎中的各收缩阵列共同执行节点的组合任务。

可选地,所述存储模块 405 还用于,若所述组合引擎为所述独立工作模式,所述聚合引擎执行完成少于第一设定数量的聚合任务后,将得到的少于所述第一设定数量的聚合特征存入所述缓存区;

所述执行模块 406 具体用于,所述组合引擎监测到所述缓存区中存储有少于第一设定数量的聚合特征,获取到所述少于第一设定数量的聚合特征,按照所述独立工作模式通过收缩阵列执行组合任务。

可选地,所述存储模块 405 还用于,若所述组合引擎为所述组合工作模式,所述聚合引擎执行完成不少于第二设定数量的聚合任务后,将得到的少于所述第二设定数量的聚合特征存入所述缓存区;

所述执行模块 406 具体用于,所述组合引擎监测到所述缓存区中存储有不

少于第二设定数量的聚合特征，获取到所述不少于第二设定数量的聚合特征，按照所述合作工作模式通过收缩阵列执行组合任务。

可选地，针对所述内存以及所述缓存区预设有访问优先级。

本说明书还提供了一种计算机可读存储介质，该存储介质存储有计算机程序，计算机程序可用于执行上述针对图卷积加速的执行方法。

本说明书还提供了图 5 所示的电子设备的示意结构图。如图 5 所述，在硬件层面，该电子设备包括处理器、内部总线、网络接口、内存以及非易失性存储器，当然还可能包括其他业务所需要的硬件。处理器从非易失性存储器中读取对应的计算机程序到内存中然后运行，以实现上述针对图卷积加速的执行方法。

当然，除了软件实现方式之外，本说明书并不排除其他实现方式，比如逻辑器件抑或软硬件结合的方式等等，也就是说以下处理流程的执行主体并不限于各个逻辑单元，也可以是硬件或逻辑器件。

在 20 世纪 90 年代，对于一个技术的改进可以很明显地区分是硬件上的改进（例如，对二极管、晶体管、开关等电路结构的改进）还是软件上的改进（对于方法流程的改进）。然而，随着技术的发展，当今的很多方法流程的改进已经可以视为硬件电路结构的直接改进。设计人员几乎都通过将改进的方法流程编程到硬件电路中来得到相应的硬件电路结构。因此，不能说一个方法流程的改进就不能用硬件实体模块来实现。例如，可编程逻辑器件（Programmable Logic Device, PLD）（例如现场可编程门阵列（Field Programmable Gate Array, FPGA））就是这样一种集成电路，其逻辑功能由用户对器件编程来确定。由设计人员自行编程来把一个数字系统“集成”在一片 PLD 上，而不需要请芯片制造厂商来设计和制作专用的集成电路芯片。而且，如今，取代手工地制作集成电路芯片，这种编程也多半改用“逻辑编译器（logic compiler）”软件来实现，它与程序开发撰写时所用的软件编译器相类似，而要编译之前的原始代码也得用特定的编程语言来撰写，此称之为硬件描述语言（Hardware Description

Language, HDL), 而 HDL 也并非仅有一种, 而是有许多种, 如 ABEL(Advanced Boolean Expression Language)、AHDL(Altera Hardware Description Language)、Confluence、CUPL(Cornell University Programming Language)、HDCal、JHDL(Java Hardware Description Language)、Lava、Lola、MyHDL、PALASM、  
 5 RHDL(Ruby Hardware Description Language)等, 目前最普遍使用的是 VHDL(Very-High-Speed Integrated Circuit Hardware Description Language)与 Verilog。本领域技术人员也应该清楚, 只需要将方法流程用上述几种硬件描述语言稍作逻辑编程并编程到集成电路中, 就可以很容易得到实现该逻辑方法流程的硬件电路。

10 控制器可以按任何适当的方式实现, 例如, 控制器可以采取例如微处理器或处理器以及存储可由该(微)处理器执行的计算机可读程序代码(例如软件或固件)的计算机可读介质、逻辑门、开关、专用集成电路(Application Specific Integrated Circuit, ASIC)、可编程逻辑控制器和嵌入微控制器的形式, 控制器的例子包括但不限于以下微控制器: ARC 625D、Atmel AT91SAM、Microchip  
 15 PIC18F26K20 以及 Silicone Labs C8051F320, 存储器控制器还可以被实现为存储器的控制逻辑的一部分。本领域技术人员也知道, 除了以纯计算机可读程序代码方式实现控制器以外, 完全可以通过将方法步骤进行逻辑编程来使得控制器以逻辑门、开关、专用集成电路、可编程逻辑控制器和嵌入微控制器等的形式来实现相同功能。因此这种控制器可以被认为是一种硬件部件, 而对其内包  
 20 括的用于实现各种功能的装置也可以视为硬件部件内的结构。或者甚至, 可以将用于实现各种功能的装置视为既可以是实现方法的软件模块又可以是硬件部件内的结构。

上述实施例阐明的系统、装置、模块或单元, 具体可以由计算机芯片或实体实现, 或者由具有某种功能的产品来实现。一种典型的实现设备为计算机。  
 25 具体的, 计算机例如可以为个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件设备、游戏控制

台、平板计算机、可穿戴设备或者这些设备中的任何设备的组合。

为了描述的方便，描述以上装置时以功能分为各种单元分别描述。当然，在实施本说明书时可以把各单元的功能在同一个或多个软件和/或硬件中实现。

本领域内的技术人员应明白，本说明书的实施例可提供为方法、系统、或  
5 计算机程序产品。因此，本说明书可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本说明书可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

本说明书是参照根据本说明书实施例的方法、设备（系统）、和计算机程  
10 序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程  
15 图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个  
20 流程和/或方框图一个方框或多个方框中指定的功能。

这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

25 在一个典型的配置中，计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器(RAM)和/或非易失性内存等形式，如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。

计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存(PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括暂存电脑可读媒体(transitory media)，如调制的数据信号和载波。

还需要说明的是，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

本领域技术人员应明白，本说明书的实施例可提供为方法、系统或计算机程序产品。因此，本说明书可采用完全硬件实施例、完全软件实施例或结合软件和硬件方面的实施例的形式。而且，本说明书可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

本说明书可以在由计算机执行的计算机可执行指令的一般上下文中描述，例如程序模块。一般地，程序模块包括执行特定任务或实现特定抽象数据类型



的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本说明书，在这些分布式计算环境中，由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中，程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

- 5       本说明书中的各个实施例均采用递进的方式描述，各个实施例之间相同相似的部分互相参见即可，每个实施例重点说明的都是与其他实施例的不同之处。尤其，对于系统实施例而言，由于其基本相似于方法实施例，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。

- 10       以上所述仅为本说明书的实施例而已，并不用于限制本说明书。对于本领域技术人员来说，本说明书可以有各种更改和变化。凡在本说明书的精神和原理之内所作的任何修改、等同替换、改进等，均应包含在本说明书的权利要求范围之内。

说明书附图

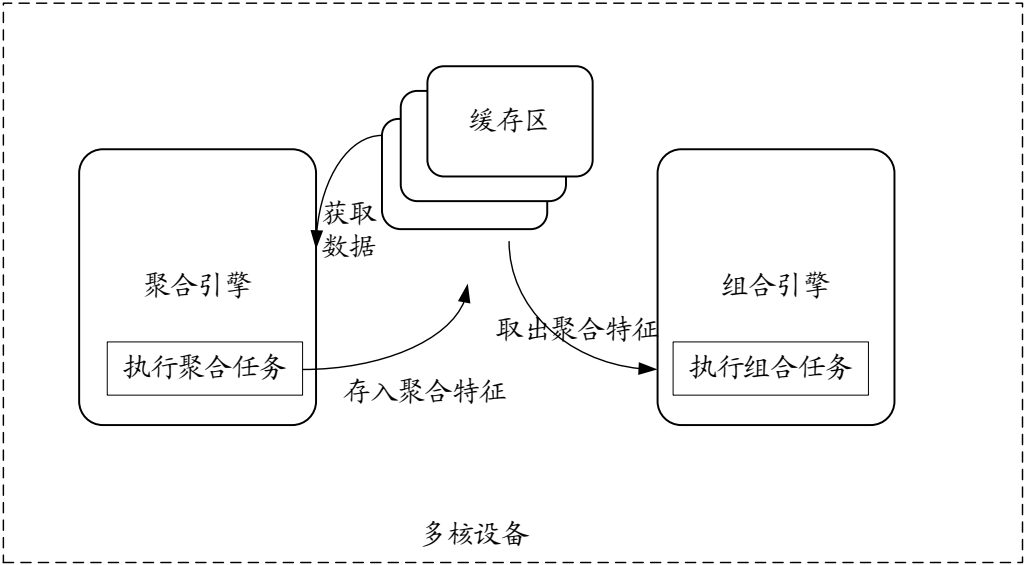


图 1

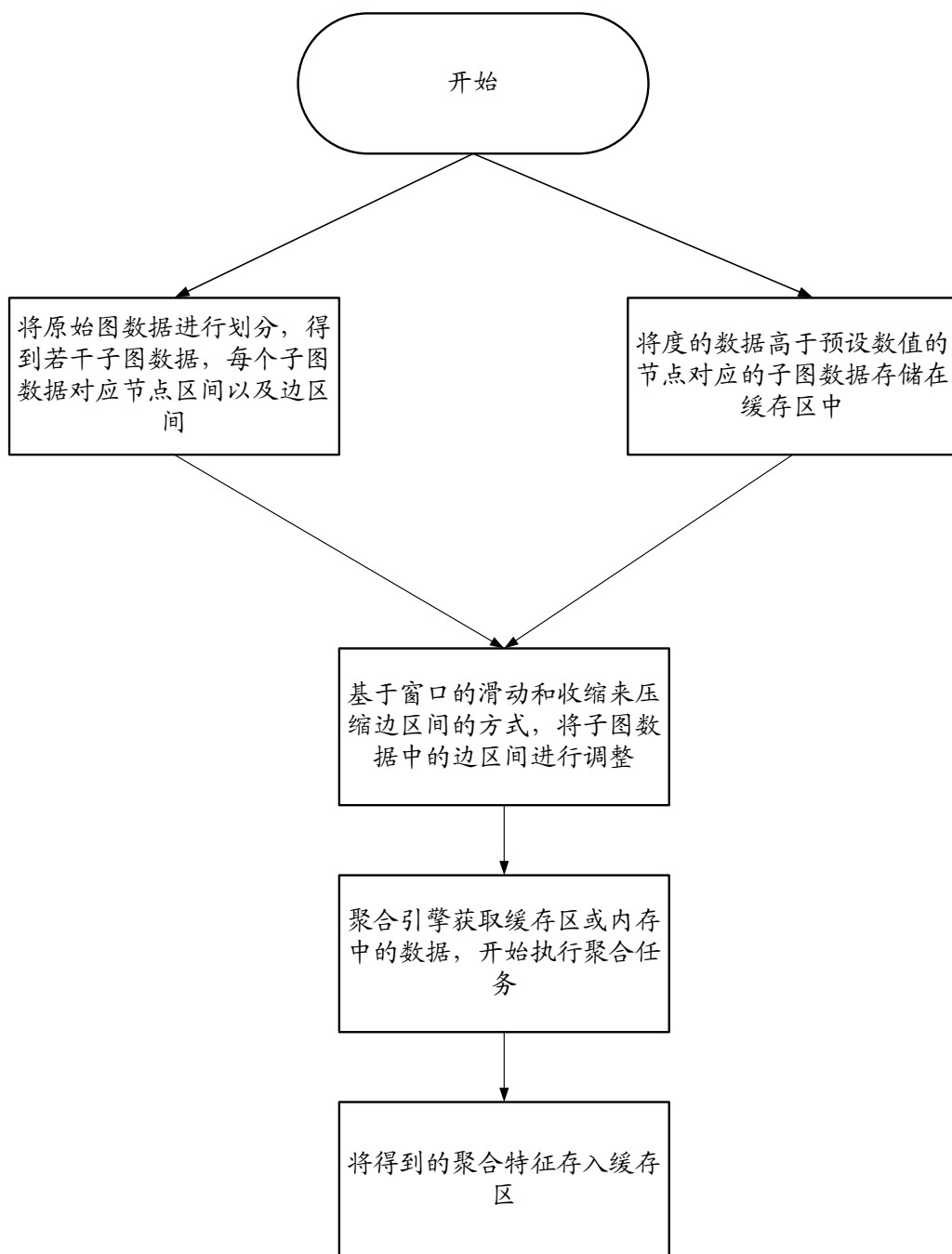


图2

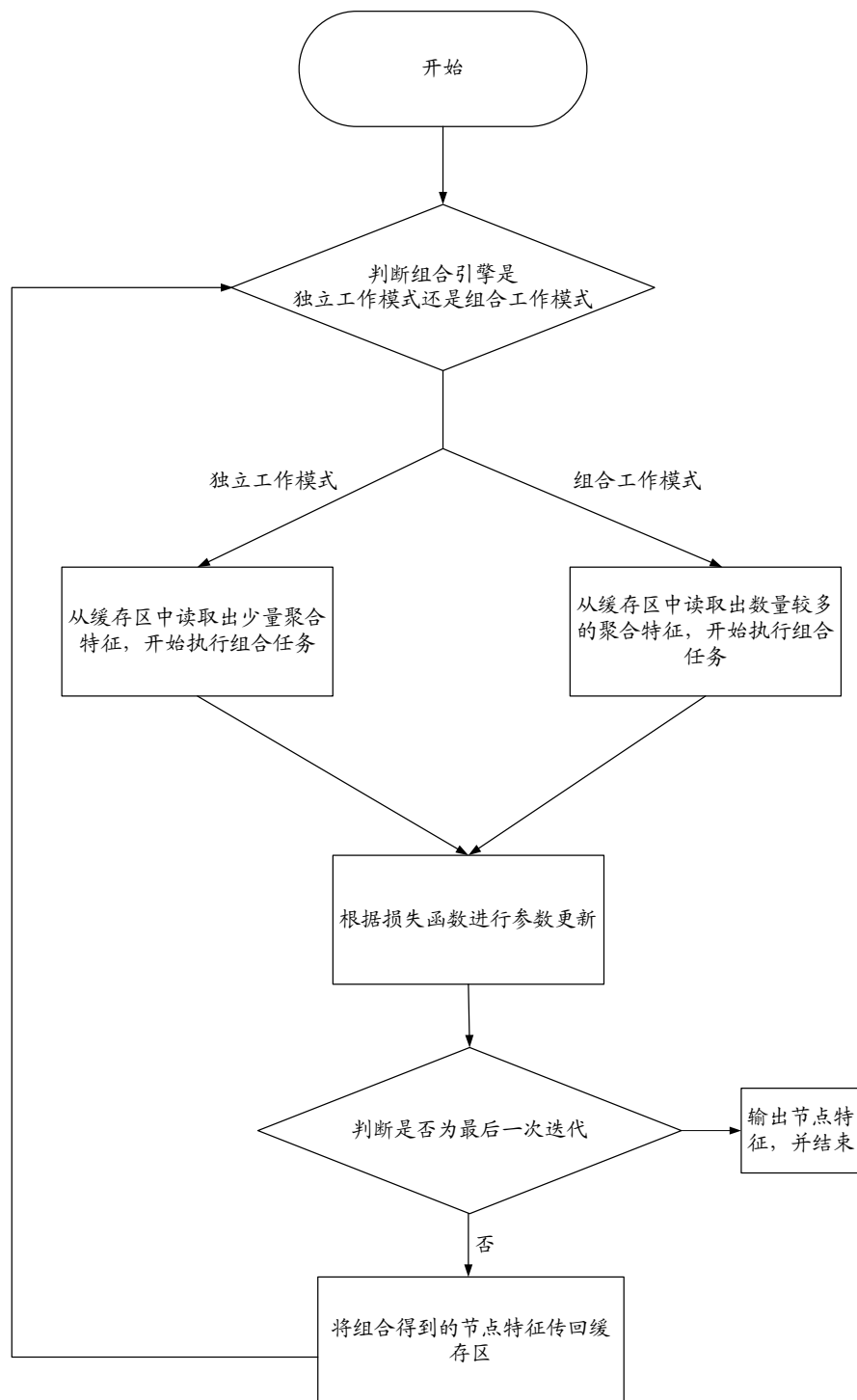


图3

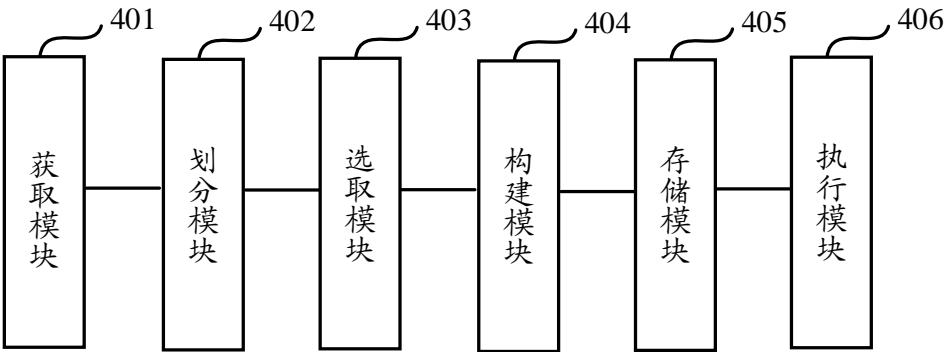


图 4

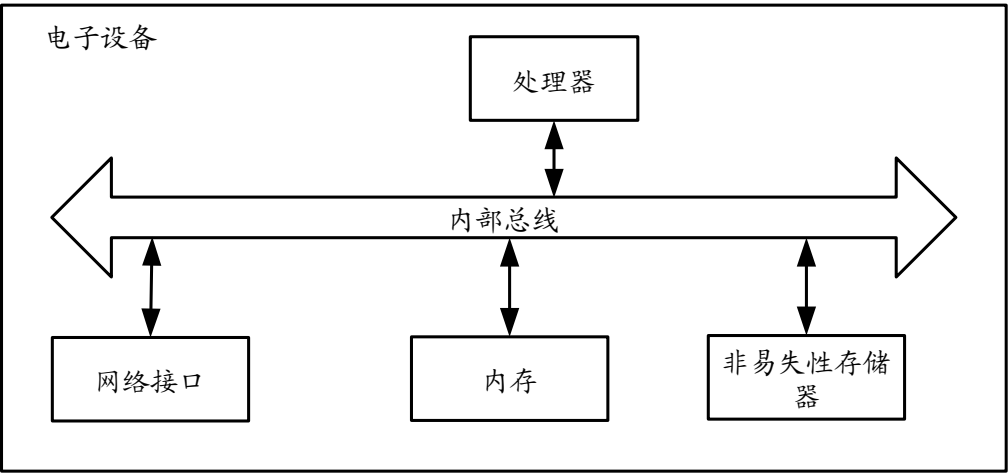


图 5