

一种矩阵计算装置、方法、系统、电路、芯片及设备

5 技术领域

本申请涉及计算机领域，尤其涉及一种矩阵计算装置、方法、系统、电路、芯片及设备。

背景技术

10 矩阵计算是人工智能、科学计算、图形计算等不同的应用场景中重要的计算类型。矩阵是一个按照长方阵列排列的元素值的集合，矩阵中的元素值可能包含有零元素值和非零元素值这两种数值，当矩阵中有大量的零元素值时，为了节省存储空间，可以仅存储矩阵中的非零元素值，即对矩阵进行压缩，存储压缩格式的矩阵。

当前技术中，针对压缩格式的矩阵的计算常用方法是，首先需要将压缩格式的矩阵进行解压缩，即将压缩格式的矩阵转换成非压缩格式的矩阵，然后再针对非压缩格式的矩阵做矩阵计算。在矩阵计算过程中，由于需要对压缩格式的矩阵进行解压缩操作，且解压缩后的数据会占用非常大的内存空间，这使得矩阵的计算速度会受限于内存的存取带宽，在内存的存取带宽一定的情况下，矩阵的计算速度无法提升，从而导致计算效能较低。

20 发明内容

本申请提供一种矩阵计算装置、方法、系统、电路、芯片及设备，能够直接对压缩格式的矩阵进行计算，不需要将压缩格式的矩阵进行解压缩，进而能够提升压缩格式的矩阵的计算效能。

第一方面，本申请实施例提供了一种矩阵计算装置，矩阵计算装置包括：向量外积处理引擎和累加器；向量外积处理引擎，用于计算N个第一列向量和N个第一行向量的向量外积，得到N个中间结果矩阵，第一列向量包括第一元素值及第一元素值的行坐标，第一行向量包括第二元素值及第二元素值的列坐标，中间结果矩阵包括第三元素值及第三元素值的位置坐标，位置坐标包括第一元素值的行坐标和第二元素值的列坐标；其中，N个第一列向量是由压缩格式的第一矩阵转换得到的，N个第一行向量是由压缩格式的第二矩阵转换得到的，N为大于或者等于1的整数；累加器，用于根据第三元素值的位置坐标的索引，将N个中间结果矩阵中位置坐标相同的第三元素值进行累加，得到结果矩阵。本申请实施例中，矩阵计算装置基于向量外积对压缩格式的第一矩阵和第二矩阵进行计算，在计算的过程中，保留第一列向量中元素值的行坐标，保留第二列向量中元素值的列坐标，然后基于位置坐标的索引，将位置坐标相同的第三元素值进行累加，从而得到两个压缩格式的矩阵进行计算的结果矩阵，相对于传统的需要将压缩格式的矩阵进行先解压缩，然后对解压缩后的矩阵进行矩阵计算的方法，本申请实施例中提供的矩阵计算装置能够有效提升压缩格式的矩阵的计算效能。

35 40 在一种可选的实现方式中，N个中间结果矩阵至少包括第一中间结果矩阵和第二中间结果矩阵，第一中间结果矩阵中第三元素值的位置坐标为第一位置坐标，第二中间结果矩阵中第三元素值的位置坐标为第二位置坐标；累加器用于按照N个中间结果矩阵的产生顺序，将第一中间结果矩阵中第三元素值按照第一位置坐标写入缓存器中的对应位置；然后再根据第

二中间结果矩阵中第三元素值的第二位置坐标,读取第二位置坐标在缓存器中对应位置的缓存值,将第二中间结果矩阵中第三元素值和缓存值进行累加,得到非压缩格式的结果矩阵。在上述可选的实现方式中,矩阵计算装置可以根据不同的应用场景输出非压缩格式的矩阵,增加了矩阵计算装置的适用场景。

5 在一种可选的实现方式中,矩阵计算装置还包括矩阵压缩单元;矩阵压缩单元用于压缩非压缩格式的结果矩阵,得到压缩格式的结果矩阵。在上述可选的实现方式中,矩阵计算装置可以根据不同的应用场景输出压缩格式的矩阵,增加了矩阵计算装置的适用场景。并且矩阵计算装置将结果矩阵进行压缩,输出压缩格式的矩阵,从而节省传输资源或便于下一步计算操作。

10 在一种可选的实现方式中,累加器还具体用于:将N个中间结果矩阵中第三元素值按照第三元素值的位置坐标进行排序,例如,按照第三元素值的行坐标进行排序,或者,按照第三元素值的列坐标进行排序;然后将排序后的N个中间结果矩阵中的位置坐标进行比对,将位置坐标相同的第三元素值相加,并删除零元素值的位置坐标,得到压缩格式的结果矩阵。在上述可选的实现方式中,可以直接得到压缩格式的矩阵,矩阵计算装置输出的压缩格式的
15 矩阵可以应用于某些后续计算需要压缩格式的矩阵的应用场景。且由于矩阵计算装置输出的是压缩格式的矩阵,可以减少后续传输矩阵的传输资源。

在一种可选的实现方式中,矩阵计算装置还包括格式转换单元;格式转换单元,用于获取第一矩阵和第二矩阵,将第一矩阵转换成N个第一列向量,并保留第一列向量中第一元素值的行坐标,将第二矩阵转换成N个第一行向量,并保留第一行向量中第一元素值的列坐标,
20 从而可以使得矩阵计算装置可以基于向量外积对两个压缩格式矩阵进行计算。

在一种可选的实现方式中,矩阵计算装置还包括格式转换单元;格式转换单元用于获取第五矩阵和第六矩阵,对第五矩阵进行格式转换,得到第一矩阵,对第六矩阵进行格式转换,得到第二矩阵,其中,第五矩阵和第六矩阵中至少有一个矩阵为非压缩格式的矩阵。在上述可选的实现方式中,矩阵计算装置可以接收非压缩格式的矩阵,然后将非压缩格式的矩阵转
25 换成压缩格式的矩阵,从而使得矩阵计算装置可以支持多种格式的矩阵计算。

在一种可选的实现方式中,矩阵计算装置还包括格式转换单元,格式转换单元还用于将第一列向量拆分为X个第二列向量,将第一行向量拆分为X个第二行向量,其中,第二列向量和第二行向量包含的元素值的精度为第二精度,第一列向量和第一行向量中包括的元素值的精度为第一精度,而且第一精度高于第二精度,X为大于或者等于2的整数;向量外积处理引擎,还用于计算X个第二列向量和X个第二行向量的向量外积,得到 X^2 个第四矩阵,第四矩阵包括第四元素值及第四元素值的位置坐标,第四元素值的位置坐标包括第一元素值的行坐标和第二元素值的列坐标,第四元素值的精度为第一精度;然后累加器还用于根据第四元素值的位置坐标的索引,将 X^2 个第四矩阵中位置坐标相同的第四元素值累加,得到中间结果矩阵,中间结果矩阵中的第三元素值的精度为第一精度。在上述可选的实现方式中,矩阵
30 计算装置可基于低精度的矩阵计算装置实现高精度的矩阵计算,从而提高矩阵计算单元的适用性。

第二方面,本申请实施例提供了一种矩阵计算方法,该方法应用于矩阵计算装置,方法包括:首先,获取第一计算指令,第一计算指令包括N个第一列向量和N个第一行向量;然后,计算N个第一列向量和N个第一行向量的向量外积,得到N个中间结果矩阵,其中,第一列向量包括第一元素值及第一元素值的行坐标,第一行向量包括第二元素值及第二元素值
40 值

的列坐标，中间结果矩阵包括第三元素值及第三元素值的位置坐标，位置坐标包括第一元素值的行坐标和第二元素值的列坐标；N个第一列向量是由压缩格式的第一矩阵转换得到的，N个第一行向量是由压缩格式的第二矩阵转换得到的，N为大于或者等于1的整数；最后，根据第三元素值的位置坐标的索引，将N个中间结果矩阵中位置坐标相同的第三元素值进行累加，得到结果矩阵。本申请实施例中，基于N个第一列向量和N个第一行向量的向量外积对第一矩阵和第二矩阵进行计算，在计算的过程中，保留第一列向量中元素值的行坐标，保留第二列向量中元素值的列坐标，然后基于位置坐标的索引，将位置坐标相同的第三元素值进行累加，从而得到两个压缩格式的矩阵进行计算的结果矩阵，相对于传统的需要将压缩格式的矩阵进行先解压缩，然后对解压缩后的矩阵进行矩阵计算的方法，本申请实施例中提供的矩阵计算装置能够有效提升压缩格式的矩阵的计算效能。

在一种可选的实现方式中，N个中间结果矩阵至少包括第一中间结果矩阵和第二中间结果矩阵，第一中间结果矩阵中第三元素值的位置坐标为第一位置坐标，第二中间结果矩阵中第三元素值的位置坐标为第二位置坐标；该方法中根据第三元素值的位置坐标的索引，将N个中间结果矩阵中位置坐标相同的第三元素值进行累加，得到结果矩阵可以具体包括：先按照N个中间结果矩阵的产生顺序，将第一中间结果矩阵中第三元素值按照第一位置坐标写入缓存器中的对应位置；然后进一步根据第二中间结果矩阵中第三元素值的第二位置坐标，读取第二位置坐标在缓存器中对应位置的缓存值，将第二中间结果矩阵中第三元素值和缓存值进行累加，得到非压缩格式的结果矩阵。在上述可选的实现方式中，矩阵计算装置可以根据不同的应用场景输出非压缩格式的矩阵，增加了矩阵计算装置的适用场景。

在一种可选的实现方式中，该方法还包括：压缩非压缩格式的结果矩阵，得到压缩格式的结果矩阵。在上述可选的实现方式中，矩阵计算装置可以根据不同的应用场景输出压缩格式的矩阵，增加了矩阵计算装置的适用场景。并且矩阵计算装置将结果矩阵进行压缩，输出压缩格式的矩阵，从而节省传输资源或便于下一步计算操作。

在一种可选的实现方式中，该方法中根据第三元素值的位置坐标的索引，将N个中间结果矩阵中位置坐标相同的第三元素值进行累加得到结果矩阵，可以具体包括：首先，将N个中间结果矩阵中第三元素值按照第三元素值的位置坐标进行排序，例如，按照第三元素值的行坐标进行排序，或者，按照第三元素值的列坐标进行排序；将排序后的N个中间结果矩阵中的位置坐标进行比对，将位置坐标相同的第三元素值相加，并删除零元素值的位置坐标，得到压缩格式的结果矩阵。在上述可选的实现方式中，可以直接得到压缩格式的矩阵，矩阵计算装置输出的压缩格式的矩阵可以应用于某些后续计算需要压缩格式的矩阵的应用场景。且由于矩阵计算装置输出的是压缩格式的矩阵，可以减少后续传输矩阵的传输资源。

在一种可选的实现方式中，在获取第一计算指令之前，该方法还包括：获取第二计算指令，第二计算指令包括第一矩阵和第二矩阵；将第一矩阵转换成N个第一列向量，并保留第一列向量中第一元素值的行坐标，将第二矩阵转换成N个第一行向量，并保留第一行向量中第一元素值的列坐标，从而可以使得矩阵计算装置可以基于向量外积对两个压缩格式矩阵进行计算。

在一种可选的实现方式中，在获取第二计算指令之前，该方法还包括：获取第三计算指令，第三计算指令包括第五矩阵和第六矩阵，其中，第五矩阵和第六矩阵中至少有一个矩阵为非压缩格式的矩阵；然后对第五矩阵进行格式转换，得到压缩格式的第一矩阵，对第六矩阵进行格式转换，得到第二矩阵。在上述可选的实现方式中，矩阵计算装置可以接收非压缩

格式的矩阵，然后将非压缩格式的矩阵转换成压缩格式的矩阵，从而使得矩阵计算装置可以支持多种格式的矩阵计算。

在一种可选的实现方式中，可以先将第一列向量拆分成 X 个第二列向量，将第一行向量拆分成 X 个第二行向量，其中，第二列向量和第二行向量包含的元素值的精度为第二精度，第一列向量和第一行向量中包括的元素值的精度为第一精度；第一精度高于第二精度， X 为大于或者等于 2 的整数；该方法中计算 N 个第一列向量和 N 个第一行向量的向量外积，得到 N 个中间结果矩阵可以包括：计算 X 个第二列向量和 X 个第二行向量的向量外积，得到 X^2 个第四矩阵，第四矩阵包括第四元素值及第四元素值的位置坐标，第四元素值的位置坐标包括第一元素值的行坐标和第二元素值的列坐标，第四元素值的精度为第一精度；再进一步的根据第四元素值的位置坐标的索引，将 X^2 个第四矩阵中位置坐标相同的第四元素值累加，得到中间结果矩阵，中间结果矩阵中的第三元素值的精度为第一精度。在上述可选的实现方式中，矩阵计算装置可基于低精度的矩阵计算装置实现高精度的矩阵计算，从而提高矩阵计算单元的适用性。

第三方面，提供一种矩阵计算电路，该矩阵计算电路用于执行上述第二方面或者第二方面的任一种可能的实现方式所提供的矩阵计算方法的操作步骤。

第四方面，提供一种矩阵计算系统，该系统包括处理器和矩阵计算装置，该处理器用于向该矩阵计算装置发送计算指令，该矩阵计算装置用于执行上述第二方面或者第二方面的任一种可能的实现方式所提供的矩阵计算方法的操作步骤。

第五方面，提供一种芯片，该芯片包括处理器，该处理器中集成有矩阵计算装置，该矩阵计算装置用于执行上述第二方面或者第二方面的任一种可能的实现方式所提供的矩阵计算方法的操作步骤。

第六方面，提供一种矩阵计算设备，该设备包括上述第四方面所提供的矩阵计算系统、或者上述第五方面所提供的芯片。

第七方面，提供一种可读存储介质，该可读存储介质中存储有指令，当该可读存储介质在设备上运行时，使得该设备执行第二方面或者第二方面的任一项可能的实现方式所提供的矩阵计算方法的操作步骤。

第八方面，提供一种计算机程序产品，当该计算机程序产品在计算机上运行时，使得该计算机执行第二方面或者第二方面的任一项可能的实现方式所提供的矩阵计算方法的操作步骤。

可以理解地，上述提供的任一种矩阵计算方法的装置、计算机存储介质或者计算机程序产品均用于执行上文所提供的对应的方法，因此，其所能达到的有益效果可参考上文所提供的对应的方法中的有益效果，此处不再赘述。

本申请在上述各方面提供的实现方式的基础上，还可以进行进一步组合以提供更多实现方式。

附图说明

图 1A 为本申请实施例中 COO 压缩格式的矩阵的示意图；

图 1B 为本申请实施例中 CSR 压缩格式的矩阵的示意图；

图 1C 为本申请实施例中 CSC 压缩格式的矩阵的示意图；

图 2 为本申请实施例中提供的一种计算设备的结构示意图；

- 图 3 为本申请实施例中提供的一种处理器的结构示意图；
图 4A 为本申请实施例中提供的一种矩阵计算装置的结构示意图；
图 4B 为本申请实施例中提供的一种向量外积处理引擎的结构示意图；
图 4C 为本申请实施例中提供的一种 MAC 运算子单元的结构示意图；
5 图 4D 为本申请实施例中提供的一种累加器的结构示意图；
图 4E 为本申请实施例中提供的一种加法器的结构示意图；
图 5A 为本申请实施例中矩阵计算装置将第一矩阵转换为 N 个第一列向量及第二矩阵转换为多个第一行向量的示意图；
图 5B 为本申请实施例中的矩阵计算装置对 N 个第一列向量和 N 个第一行向量进行外积
10 计算，得到 N 个中间结果矩阵的示意图；
图 5C 为本申请实施例中的第一中间结果矩阵中的元素值及位置坐标的示意图；
图 5D 为本申请实施例中的第二中间结果矩阵中的元素值及位置坐标的示意图；
图 5E 为本申请实施例中矩阵计算装置将 N 个中间结果矩阵中位置坐标相同的第三元素
值进行累加得到结果矩阵的示意图；
15 图 6 为本申请实施例中将 N 个中间结果矩阵中位置坐标相同的第三元素值进行累加得到
结果矩阵的一种实现方式的示意图；
图 7 为本申请实施例中提供的另一种矩阵计算装置的结构示意图；
图 8 为本申请实施例中将 N 个中间结果矩阵中位置坐标相同的第三元素值进行累加得到
结果矩阵的另一种实现方式的示意图；
20 图 9 为本申请实施例中格式转换单元对非压缩格式的矩阵转换为压缩格式的矩阵的示
意图；
图 10 为本申请实施例中将第一精度的整型数值分解成多个第二精度的整型数值的示
意图；
图 11 为本申请实施例中将第一精度的第一列向量分解成多个第二精度的第二列向量及
25 第一精度的第一行向量分解成多个第二精度的第二行向量的示意图；
图 12 为本申请实施例中矩阵计算装置对第二行向量和第二列向量进行向量外积计算得
到第四矩阵的示意图；
图 13 为本申请实施例中一种矩阵计算方法的一个实施例的步骤流程示意图；
图 14 为本申请实施例中一种矩阵计算方法的另一个实施例的步骤流程示意图。

30

具体实施方式

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行描述。本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象，而不必然用于描述特定的顺序或先后次序。

35

为了更好的理解本申请，首先对本申请中涉及的相关词语进行说明。
矩阵 (matrix): 一个维度为 $m \times n$ 矩阵是一个由 m 行 (row) n 列 (column) 的元素排
列成的矩形阵列。示例性的，矩阵A如下式 (1) 所示，矩阵B如下式 (2) 所示。

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \text{ 式 (1)。 } B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix}, \text{ 式 (2)。}$$

矩阵的加减法：维度相同的矩阵之间可以相互加减，具体是对每个位置上的元素做加减法。例如，矩阵A与矩阵B均为维度为 $m \times n$ 矩阵，矩阵A和矩阵B相加得到矩阵，矩阵C如下式(3)所示。

$$C = A + B = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \cdots & a_{mn} + b_{mn} \end{bmatrix}, \text{ 式 (3)。}$$

矩阵的乘法：两个矩阵的乘法仅当第一个矩阵A的列数(column)和另一个矩阵的行数(row)相等时才能相乘。例如，矩阵A是维度为 $m \times n$ 的矩阵，矩阵B是维度为 $n \times p$ 的矩阵，矩阵A和矩阵B的乘积是一个 $m \times p$ 矩阵，该 $m \times p$ 矩阵中的一个元素如下式(4)所示：

$$[ab]_{i,j} = a_{i,1}b_{1,j} + a_{i,2}b_{2,j} + \cdots + a_{i,n}b_{n,j} = \sum_{r=1}^n a_{i,r}b_{r,j}, \text{ 式 (4)，其中， } 1 \leq i \leq m, 1 \leq j \leq p.$$

行向量(row vector)是一个维度为 $1 \times m$ 的矩阵，其中， m 为正整数。例如：一个行向量如下式(5)所示：

$$X = [x_1 \ x_2 \ \cdots \ x_m], \text{ 式 (5)。}$$

列向量(column vector)是一个维度为 $m \times 1$ 的矩阵， m 为正整数。例如：一个列向量如下式(6)所示：

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \text{ 式 (6)。}$$

向量外积(vector outer product)：两个向量的张量积，张量积为一矩阵。例如，给定一个维度为 $m \times 1$ 的列向量U和维度为 $1 \times n$ 的行向量V，向量U和向量V的外积， $U \otimes V$ 被定义为维度为 $m \times n$ 的矩阵D，矩阵D如下式(7)所示：

$$U \otimes V = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \otimes [v_1 \ v_2 \ v_3] = \begin{bmatrix} v_1 u_1 & v_2 u_1 & v_3 u_1 \\ v_1 u_2 & v_2 u_2 & v_3 u_2 \\ v_1 u_3 & v_2 u_3 & v_3 u_3 \\ v_1 u_4 & v_2 u_4 & v_3 u_4 \end{bmatrix}, \text{ 式 (7)。}$$

压缩格式的矩阵：当一个矩阵中包含了零元素值和非零元素值时，通常情况下，为了节省存储空间，可以按照一定格式存储矩阵中的非零元素值，而不存储零元素值，这个过程即对矩阵进行压缩，压缩存储后的矩阵称为压缩格式的矩阵。对矩阵进行压缩的方法包括但不限于坐标表示(coordinate, COO)、行压缩(compressed sparse row, CSR)、列压缩(compressed sparse column, CSC)等。

下面对分别对COO、CSR和CSC这三种压缩方法进行示例性说明。

COO：通过三元组来表示矩阵。三元组中包括三个值，三个值分别是行号、列号和元素值。其中，行号和列号用于标识元素值的位置。例如，三元组为(行号，列号，元素值)，

或者，三元组为（元素值，行号，列号）等，具体的并不限定三元组中三个值的排列顺序。举例来说，请参阅图1A，图1A中示出了一个维度为 4×4 的矩阵Y，该矩阵中包括零元素值和非零元素值，其中，非零元素值为：1，2，3，4，5，6，7，8，9。如非零元素值“1”所在的位置是第0行第0列，三元组表示为（0，0，1）。非零元素值“2”所在的位置是第0行第1列，三元组表示为（0，1，2）。非零元素值“3”所在的位置是第1行第1列，三元组表示为（1，1，3）。此处不对每个元素值进行一一赘述说明。例如，压缩格式的矩阵的三元组形式为：（0，0，1），（0，1，2），（1，1，3），（1，2，4），（2，0，5），（2，2，6），（2，3，7），（3，1，8），（3，3，9）。

可选地，压缩格式的矩阵Y可以表示为如下式（8）所示。

行坐标=([0,0,1,1,2,2,2,3,3])
列坐标= ([0,1,1,2,0,2,3,1,3])
元素值= ([1,2,3,4,5,6,7,8,9])，式（8）。

CSR：通过三类数据来表示矩阵，三类数据分别是元素值、列号以及行偏移值。其中，CSR中的元素值和列号与上述COO方式中的元素值和列号的表示方式类似。CSR与COO方式不同的是，行偏移表示某一行的第一个元素在所有元素值中的起始偏移位置。请参阅图1B所示，首先，将图1B中所示的矩阵Y中的非零元素值按照行进行排列，得到所有的元素值为：1，2，3，4，5，6，7，8，9。第一行中的第一个非零元素值是“1”，元素值“1”在所有元素值中的偏移是“0”。同理，第二行中的第一个非零元素值是“3”，元素值“3”在所有元素值中的偏移是“2”。第三行中的第一个非零元素值是“5”，元素值“5”在所有元素值中的偏移是“4”，第四行中的第一个非零元素值是“8”，元素值“8”在所有元素值中的偏移是“7”。最后在行偏移所在行的最后位置补上矩阵中非零元素值的总个数（如“9”）。

压缩格式的矩阵Y可以表示为如下式（9）所示。

行偏移=([0,2,4,7,9])
列坐标=([0,1,1,2,0,2,3,1,3])
元素值=([1,2,3,4,5,6,7,8,9])，式（9）。

CSC：通过三类数据来表示矩阵，三类数据分别是元素值、行号以及列偏移值。其中，CSR中的元素值和行号与上述COO方式中的元素值和行号的表示方式类似。CSC与COO方式不同的是，列偏移表示某一列的第一个元素在所有元素值中的起始偏移位置。请参阅图1C所示，首先，将图1C中所示的矩阵Y中的非零元素值按照列进行排列，得到所有的元素值为：1，5，2，3，8，4，6，7，9。第一列中的第一个非零元素值是“1”，元素值“1”在所有元素值中的偏移是“0”。同理，第二列中的第一个非零元素值是“5”，元素值“5”在所有元素值中的偏移是“2”。第三列中的第一个非零元素值是“2”，元素值“2”在所有元素值中的偏移是“5”，第四列中的第一个非零元素值是“3”，元素值“3”在所有元素值中的偏移是“7”。最后在列偏移所在行的最后位置补上矩阵中非零元素值的总个数（如“9”）。

压缩格式的矩阵Y可以表示为如下式（10）所示。

列偏移= ([0,2,5,7,9])
行坐标= ([0,0,1,1,2,2,2,3,3])
元素值= ([1,5,2,3,8,4,6,7,9])，式（10）。

通过对上述三种矩阵压缩方法的说明可知，COO压缩格式的矩阵中每个元素值具有对应

的行坐标（行号）和列坐标（列号）。CSR压缩格式的矩阵中每个元素值都会有对应的列坐标。CSC压缩格式的矩阵中每个元素值都会有对应的行坐标。

非压缩格式的矩阵：请参阅图1A中示出的矩阵Y，非压缩格式矩阵包括零元素值和非零元素值。需要说明的是，通常情况下，将压缩格式的矩阵也称为稀疏矩阵，而将非压缩格式的矩阵也可以称为稠密矩阵。

数值类型：在计算机领域中，数值类型包括整型和浮点型。整型主要用于表示整数，浮点型主要用于表示小数。其中，整型的精度包括 int2、int4、int8、int16、int32 等精度。其中，int 用于表示整型函数，int 后加上数字就代表 2 进制值域有多少位（比特 bit），1 个位（bit）是 0 或者 1。如 int4 的 2 进制值域是 4 位（0000-1111），换算到 10 进制为（-8，7）。同理 int8 的 2 进制值域为（ -2^7 ， 2^7-1 ）。计算机储存容量单位 1 字节是 8 位，所以对

应的 int8 有 1 字节，int16 有 2 字节。int16 的 2 进制值域占 2 个字节，换算到 10 进制为（-32768，32767）。int32 的 2 进制值域占 4 个字节，换算到 10 进制为（-2147483648，2147483647）。

整型矩阵是指以整型数值作为元素的矩阵。比如，m 行 n 列的整型矩阵包括 $m \times n$ 个元素，这 $m \times n$ 个元素为整型数值。该整型数值可以为 int2、int4、int8、int16 或 int32 等精度，如整型矩阵也可以包括不同整型格式的矩阵，比如，包括 int8 格式的整型数值的矩阵、包括 int16 格式的整型数值的矩阵、以及包括 int32 格式的整型数值的矩阵等。

浮点数（float point，FP）主要用于表示小数，通常由三部分组成，即符号（sign）位、指数（exponent）位和尾数（mantisa）位，该指数位也可以称为阶码位。其中，符号位可以为 1 比特（bit），阶码位和尾数位可以为多个比特（bits）。浮点数通常可以包括多种格式（format），如 IEEE754 标准中的半精度浮点数、单精度浮点数和双精度浮点数等。其中，半精度浮点数（half-precision floating-point）在计算机存储器中占用 16bits（即占用 2 个字节），也可以简称为 FP16。半精度浮点数可表示的数值的绝对值范围大约为 $[6.10 \times 10^{-5}, 6.55 \times 10^4]$ 。单精度浮点数（single-precision floating-point）在计算机存储器中占用 32 bits（即 4 个字节），也可以简称为 FP32。单精度浮点数可表示的数值的绝对值范围大约为 $[1.18 \times 10^{-38}, 3.40 \times 10^{38}]$ 。双精度浮点数（double precision floating point）在计算机存储器中占用 64 bits（即占用 8 个字节），也可以简称为 FP64。双精度浮点数可以表示十进制的 15 位或 16 位有效数字，可表示的数值的绝对值范围大约为 $[2.23 \times 10^{-308}, 1.80 \times 10^{308}]$ 。

下面表 1 中示出了上述三种浮点数的一种存储格式。其中，在 FP16 占用的 16 bits 中，符号位占用 1 bit，阶码占用 5 bits，尾数位占用 10 bits。在 FP32 占用的 32 bits 中，符号位占用 1 bit，阶码占用 8 bits，尾数位占用 23 bits。在 FP64 占用的 64 bits 中，符号位占用 1 bit，阶码占用 11 bits，尾数位占用 52 bits。

表 1

	符号位	指数位（阶码）	尾数位
FP16	1 bit	5 bits	10 bits
FP32	1 bit	8 bits	23 bits
FP64	1 bit	11 bits	52 bits

浮点数矩阵可以是指以浮点数作为元素的矩阵。比如， m 行 n 列的浮点数矩阵包括 $m \times n$ 个元素，这 $m \times n$ 个元素可以为浮点数。与浮点数类似，浮点数矩阵也可以包括不同浮点数格式的矩阵，比如，包括 FP16 格式的浮点数的矩阵，包括 FP32 格式的浮点数的矩阵，以及包括 FP64 格式的浮点数的矩阵等。

5

图 2 为本实施例提供的一种计算设备的结构示意图，该计算设备可以是终端、网络设备或者服务器等具有计算能力的设备。参见图 2，该计算设备可以包括存储器 201、处理器 202、通信接口 203 和总线 204，存储器 201、处理器 202 以及通信接口 203 通过总线 204 相互连接。

10 其中，存储器 201 可用于存储数据、软件程序以及模块，主要包括存储程序区和存储数据区，存储程序区可存储操作系统，至少一个功能所需的软件应用和中间层软件等，存储数据区可存储该设备的使用时所创建的数据等。比如，该操作系统可以包括 Linux 操作系统、Unix 操作系统或者 Window 操作系统等；该至少一个功能所需的软件应用可以包括人工智能（artificial intelligence）相关的应用，高性能计算（high performance computing, HPC）相关的应用，深度学习（deep learning）相关的应用或者科学计算相关的应用等，该中间层软件可以包括线性代数库函数等。在一种可能的示例性中，存储器 201 包括但不限于静态随机存储器（static RAM, SRAM）、动态随机存储器（dynamic RAM, DRAM），同步动态随机存储器（synchronous DRAM, SDRAM）或者高速随机存取存储器等。进一步的，存储器 201 还可以包括其他非易失性存储器，
20 例如至少一个磁盘存储器件、闪存器件、或其他易失性固态存储器件。

另外，处理器 202 用于对该计算设备的操作进行控制管理，比如通过运行或执行存储在存储器 201 内的软件程序和/或模块，以及调用存储在存储器 201 内的数据，执行该计算设备的各种功能和处理数据。在一种可能的示例性中，处理器 202 包括但不限于中央处理单元（central processing unit, CPU）、网络处理单元（network processing unit, NPU）、图形处理器（graphics processing unit, GPU）、应用专用集成电路（application specific integrated circuit, ASIC），现场可编程门阵列（field programmable gate array, FPGA）或者其他可编程逻辑器件，晶体管逻辑器件，逻辑电路或者其任意组合。其可以实现或执行结合本申请公开内容所描述的各种示例性的逻辑方框，模块和电路。处理器 202 也可以是实现计算功能的组合，例如包含一个或多个
30 个微处理器组合，数字信号处理器和微处理器的组合等。

通信接口 203 用于实现该计算设备与外部设备进行通信。其中，通信接口 203 可以包括输入接口和输出接口。该输入接口可用于获取下述实施例中的压缩格式的第一矩阵和第二矩阵。在一些可行的实施例中，该输入接口可以只有一个输入接口，也可以有多个输入接口。该输出接口可用于输出下述实施例中的结果矩阵。在一些可行的实施例中，该结果矩阵可以由处理器直接输出，也可以先被存储于存储器中，然后经存储器输出。在另一些可行的实施例中，可以只有一个输出接口，也可以有多个输出接口。
35

总线 204 可以是外设部件互连标准（Peripheral Component Interconnect Express, PCIe）总线，或者扩展工业标准结构（extended industry standard architecture, EISA）总线等。总线 204 可以分为地址总线、数据总线、控制总线等。
40

为便于表示，图 2 中仅用一条粗线表示，但并不表示仅有一根总线或一种类型的总线。

在本实施例中，处理器 202 可以包括矩阵计算装置，该矩阵计算装置可以是 ASIC、FPGA 或者逻辑电路等，当然该装置也可以通过软件实现，本申请实施例对此不作具体限制。该矩阵计算装置可以用于计算人工智能、科学计算、图形计算等涉及的矩阵计算。

进一步的，该处理器 202 还可以包括 CPU、GPU 或 NPU 等其他处理单元中的一个或者多个。如图 3 所示，以该处理器 202 包括 CPU 1 和矩阵计算装置 2 为例，矩阵计算装置 2 可以与 CPU 1 集成在一起（比如，矩阵计算装置 2 集成在 CPU 1 所在的 SoC 的内部），也可以与 CPU 1 并列单独设置（比如，矩阵计算装置 2 以 PCIe 卡的形式设置）。具体如图 3 中的（a）和图 3 中的（b）所示。进一步的，该 CPU 1 中还可以包括控制器（controller）11、一个或者多个算术逻辑单元（arithmetic logic unit, ALU）12、缓存器（cache）13 和内存管理单元（memory management unit, MMU）14 等。图 3 中以存储器 201 为动态随机存储器 DRAM 为例进行说明。

本申请实施例中，矩阵计算装置能够对压缩格式的矩阵进行计算，矩阵计算装置在计算两个压缩格式的矩阵乘法时，首先获取由其中一个矩阵转换的 N 个第一列向量，该第一列向量会包含第一元素值及第一元素值的行坐标，获取由另一个矩阵转换的 N 个第一行向量，该第一行向量中会包含第二元素值及第二元素值的列坐标。矩阵计算装置计算 N 个第一列向量和 N 个第一行向量的向量外积，从而得到 N 个中间结果矩阵，中间结果矩阵中会包括元素值及元素值对应的位置坐标，位置坐标包括第一元素值的行坐标和所述第二元素值的列坐标。矩阵计算装置能够根据位置坐标的索引，将 N 个中间结果矩阵中位置坐标相同的第三元素值进行累加，然后得到结果矩阵。本申请实施例中，矩阵计算装置基于向量外积对第一矩阵和第二矩阵进行计算，在计算的过程中，保留第一列向量中元素值的行坐标，保留第二列向量中元素值的列坐标，然后基于位置坐标的索引，将位置坐标相同的第三元素值进行累加，从而得到两个压缩格式的矩阵进行计算的结果矩阵，相对于传统的需要将压缩格式的矩阵先解压缩，然后对解压缩后的矩阵进行矩阵计算的方法，本申请实施例中提供的矩阵计算装置能够有效提升压缩格式的矩阵的计算效能。

本申请实施例提供了一种矩阵计算装置。请参阅图 4A 所示，该矩阵计算装置包括向量外积处理引擎 401 和累加器 402。可选地，矩阵计算装置包括第一缓存器 403，向量外积处理引擎 401、累加器 402 和第一缓存器 403 依次连接。第一缓存器 403 可以是矩阵计算装置内的第一缓存器 403（如寄存器），或者，第一缓存器 403 也可以是图 3 中示出的中央处理单元 1 中的缓存器 13。可选地，矩阵计算装置还包括格式转换单元 405 和第二缓存器 406。格式转换单元 405 的功能可以通过上述图 3 中的中央处理单元 1 实现，或者，格式转换单元 405 的功能可以由矩阵计算装置内的逻辑电路实现。第二缓存器 406 可以是矩阵计算装置内的寄存器，或者，第二缓存器 406 可以是图 3 中示出的中央处理单元 1 中的缓存器 13。

格式转换单元 405，用于将压缩格式的第一矩阵转换为 N 个第一列向量，将压缩格式的第二矩阵转换为 N 个第一行向量，第一列向量包括第一元素值及第一元素值的行坐标，第一行向量包括第二元素值及第二元素值的列坐标。

可选地，下面对向量外积处理引擎 401 的结构进行说明。请参阅图 4B 所示，图 4B 为向

量外积处理引擎 401 的结构示意图。向量外积处理引擎 401 多个处理单元 (processing element, PE) 4011, 多个 PE 4011 为一个二维阵列, 每个 PE 4011 包括乘积累加 (multiply accumulate, MAC) 运算子单元 40110 和坐标合并子单元 40111。每个 PE 接收两组输入数据并输出一组数据。每组输入数据包含一个数值 (value) 与一个坐标 (index)。一组输入数据包含第一元素值及第一元素值对应的行坐标, 另一组输入数据包括第二元素值及第二元素值对应的列坐标。

每个 PE 具有两个作用, 其中一个作用为, 每个 PE 4011 接收两组输入数据, 根据该组输入数据输出一组输出数据。输出数据包括第三元素值及第三元素值的位置坐标, 其中第三元素值是对第一元素值和第二元素值进行乘法运算得到的。位置坐标是对第一行坐标和第一列坐标进行合并后得到的。例如, 以 PE 阵列中第 0 行第 0 列的 PE 为例, 一组输入数据为第一行向量中的第一元素值 (如 a_0) 及第一元素值对应的行坐标 (如 i_0)。另一组输入数据为第一列向量中的第二元素值 (如 b_0) 及第二元素值对应的列坐标 (如 j_0)。MAC 运算子单元 40110 用于接收第一元素值和第二元素值, 对第一元素值和第二元素值进行乘法计算, 然后输出第一元素值与第二元素值的乘积 (即第三元素值)。坐标合并子单元 40111 用于接收第一元素值的行坐标和第二元素值的列坐标, 对该行坐标和列坐标进行合并, 得到位置坐标。一个 PE 输出的数据包括第三坐标值和该第三坐标值对应的位置坐标。

PE 的另一个作用为, 一个 PE 需要向行方向上的下一个 PE 传输第一元素值及该第一元素值对应的列坐标, 并且该 PE 需要向列方向上的下一个 PE 传输第二元素值及该第二元素值对应的行坐标。例如, 在第一个时钟周期后, PE 阵列中的第一个 PE (即第 0 行第 0 列的 PE) 将 a_0 和 i_0 传往行方向上的下一个 PE (如第 0 行第 1 列的 PE), 并将与 b_0 和 j_0 传往列方向上的下一个 PE (即第 1 行第 0 列的 PE)。可选地, 每个 PE 的数据传输方式可以是, 依每个时钟周期往下一级的 PE 传输数据, 或者, 在一个时钟周期内, 第一个 PE 将 a_0 和 i_0 传往行方向上的下一个 PE (即第 0 行第 1 列的 PE), 直到传输到该行 (即第 0 行) 中的最后一个 PE, 及将 b_0 和 j_0 传输到该列 (第 0 列) 中的最后一个 PE。本示例中, 每个时钟单元内 PE 传递的级数可依实际需求设计, 具体的并不限定。

可选地, 下面对 PE 中的 MAC 运算子单元的结构进行说明。请参阅图 4C 所示, 图 4C 为 MAC 运算子单元的结构示意图。每个 PE 中的 MAC 运算子单元包括符号子单元 40112, 指数子单元 40113, 整数子单元 40114 和精度格式转换子单元 40115。符号子单元 40112, 指数子单元 40113 和整数子单元 40114 均与精度格式转换子单元 40115 连接。其中, 符号子单元 40112 用于处理输入数值的正负号。指数子单元 40113 用于处理两个浮点数相乘时的小数点位移计算。整数子单元 40114 用于计算两个整数的相乘计算。精度格式转换子单元 40115 用于输出一个符合规范的数值格式 (如 FP16、FP32、FP64 等浮点数格式或如 int32、int16、int8 等整数格式)。本示例中的, MAC 运算子单元为支持浮点数格式和整数格式的通用单元, 增强了矩阵计算装置的适用性。

可选地, 下面对矩阵计算装置中的累加器 402 的结构进行说明。请参阅图 4D 所示, 图 4D 为累加器 402 的结构示意图。累加器 402 包括多个累加 (Accumulator, ACC) 处理单元 4021, 即累加器 402 包括有多个 ACC 处理单元 4021 组成的 ACC 处理单元阵列。每个 ACC 处理单元 4021 接收向量外积处理引擎 401 输出的一组数据, 该组数据包括一个第三元素值及该第三元素值对应的位置坐标。每个 ACC 处理单元 4021 包括一个加法器 40210 和一个数据获取子单元 40211。其中, 数据获取子单元 40211 用于接收位置坐标, 并将该位置坐标输出

至第一缓存器 403, 从而使得加法器 40210 从第一缓存器 403 中获取该位置坐标对应的缓存值。加法器 40210 用于将第一缓存器 403 输出的缓存值与该第三元素值累加。例如, 一个 ACC 处理单元 4021 接收向量外积处理引擎 401 输出的中间结果矩阵 C_0 中的一个“第三元素值 0” (如 c_{00}) 和该 c_{00} 对应的位置坐标 (如 i_0, j_0), ACC 处理单元 4021 先将该“第三元素值 0”按照位置坐标 (如 1, 1) 写入第一缓存器 403, 然后, 该 ACC 处理单元 4021 接收向量外积处理引擎 401 输出的中间结果矩阵 C_1 中的“第三元素值 1” (如 c_{10}) 及该 c_{10} 对应的位置坐标 (如 1, 1) 时, ACC 处理单元 4021 接收第一缓存器 403 输出的该位置坐标对应的缓存值 (如 c_{00}), 然后, 加法器 40210 用于将 c_{00} 和 c_{10} 相加, 并将 c_{00} 和 c_{10} 的累加值输出到第一缓存器 403。

可选地, 请参阅图 4E 所示, 加法器 40210 还具体包括符号单元 40215、指数单元 40216、整数单元 40217 和精度格式转换单元 40218。符号单元 40215、指数单元 40216、整数单元 40217 均和精度格式转换单元 40218 连接。其中, 符号单元 40215 用于处理输入数值的正负号。指数单元 40216 用于处理两个浮点数相乘时的小数点位移计算。整数单元 40217 用于计算两个整数的乘法运算。精度格式转换单元 40218 用于输出一个符合规范的数值格式 (如 FP16、FP32、FP64 等浮点数值格式或如 int32、int16、int8 等整数格式)。本示例中的, 加法器 40210 为支持浮点数值格式和整数格式的通用单元, 增强了矩阵计算装置的适用性。

下面对格式转换单元 405 的具体作用进行说明。格式转换单元 405, 用于将压缩格式的第一矩阵转换为 N 个第一列向量, 将压缩格式的第二矩阵转换为 N 个第一行向量, 第一列向量包括第一元素值及第一元素值的行坐标, 第一行向量包括第二元素值及第二元素值的列坐标。第一矩阵的维度为 $M \times N$, 第二矩阵的维度为 $N \times K$, 其中, M、N、K 为大于或者等于 1 的整数。

参阅图 5A 所示, 为了方便说明, 本示例中, M、N、K 均以 4 为例进行说明, 即第一矩阵和第二矩阵的维度均为 4×4 。第一矩阵以矩阵 A 为例, 第二矩阵以矩阵 B 为例进行说明。第一矩阵和第二矩阵的压缩格式以 COO 为例进行说明。

格式转换单元 405, 用于将矩阵 A 按照列进行拆分, 将矩阵 A 拆分成 4 个第一列向量, 该 4 个第一列向量分别为 A_0, A_1, A_2 和 A_3 。格式转换单元 405 将矩阵 B 按照行进行拆分, 将矩阵 B 拆分成 4 个第一行向量, 该 4 个第一行向量分别为 B_0, B_1, B_2 和 B_3 。示例性的, 第一列向量以 A_0 为例进行说明, A_0 为 $[a_0, a_1, a_2, a_3]^T$, 其中, A_0 中的 a_0, a_1, a_2, a_3 均为元素值。每个元素值具有对应的行坐标。例如, a_0 的行坐标为 i_0 , a_1 的行坐标为 i_1 , a_2 的行坐标为 i_2 , a_3 的行坐标为 i_3 。第一行向量以 B_0 为例进行说明。 B_0 为 $[b_0, b_1, b_2, b_3]$, 其中, B_0 中的 b_0, b_1, b_2, b_3 均为元素值。 B_0 中的每个元素值都具有对应的列坐标。例如, b_0 的列坐标为 j_0 , b_1 的列坐标为 j_1 , b_2 的列坐标为 j_2 , b_3 的列坐标为 j_3 。应理解, 矩阵 A 是按照 N 个列进行拆分的, 拆分后的第一列向量中每个元素值只保留元素值的行坐标即可。而矩阵 B 是按照 N 个行进行拆分的, 拆分后的第一行向量中的每个元素值只保留元素值的列坐标即可。同理, A_1 为 $[c_0, c_1, c_2, c_3]^T$, 其中, A_1 中的 c_0, c_1, c_2, c_3 均为元素值。每个元素值具有对应的行坐标。例如, c_0 的行坐标为 k_0 , c_1 的行坐标为 k_1 , c_2 的行坐标为 k_2 , c_3 的行坐标为 k_3 。 B_1 为 $[d_0, d_1, d_2, d_3]$, 其中, B_1 中的 d_0, d_1, d_2, d_3 均为元素值。 B_1 中的每个元素值都具有对应的列坐标。例如, d_0 的列坐标为 l_0 , d_1 的列坐标为 l_1 , d_2 的列坐标为 l_2 , d_3 的列坐标为 l_3 。

举例来说, 压缩格式的第一矩阵以 COO 格式表示为: (0, 0, 1), (1, 0, 2), (1, 1, 3), (1, 2, 4), (2, 0, 5), (2, 2, 6), (2, 3, 7), (3, 1, 8), (3, 3, 9), (3, 0, 6)。格式转换单元 405 将压缩格式的第一矩阵按照列进行拆分, 其中, (0, 0, 1), (1, 0, 2), (2,

0, 5) 和 (3, 0, 6) 中的元素值是同一列的元素值, 即都是第 0 列的元素值。应理解, 格式转换单元 405 在对矩阵 A 按照列进行拆分时, 将元素值 “1”, “2”, “5” 和 “6” 作为 A₀ 向量中的元素值, 元素值 “1”, “2”, “5” 和 “6” 都是同一列中的元素值, 那么仅保留各元素值的行坐标即可。在这个例子中, 向量 A₀ 包括 4 个元素值, 该 4 个元素值分别为 “1”, “2”, “5” 和 “6”, 且元素值 “1” 的行坐标为 “0”, 同理, 元素值 “2” 的行坐标为 “1”, 元素值 “5” 的行坐标为 “2”, 元素值 “6” 的行坐标为 “3”。

再如, 压缩格式的第二矩阵以 C00 格式表示为: (0, 0, 1), (1, 0, 2), (2, 1, 3), (1, 2, 4), (2, 0, 5), (2, 2, 6), (2, 3, 7), (3, 1, 8), (3, 3, 9), (3, 0, 6)。格式转换单元 405 将第二矩阵按照行坐标进行拆分, 即将相同行的元素值作为同一个向量内的元素值。例如, (2, 1, 3), (2, 0, 5), (2, 2, 6) 和 (2, 3, 7) 这 4 个三元组中的行坐标相同, 那么将这 4 个三元组中的元素值拆分到一个行向量 (如行向量 B₀) 中, 元素值 “3”、“5”、“6” 和 “7” 作为第一行向量 B₀ 中的元素值, 且第一行向量 B₀ 中的每个元素值具有对应的列坐标。如元素值 “3” 的列坐标为 “1”, 元素值 “5” 的列坐标为 “0”, 元素值 “6” 的列坐标为 “2”, 元素值 “7” 的列坐标为 “3”。需要说明的是, 上述对于 C00 格式中的具体数值仅是为了方便说明而举的例子, 并不造成对本申请的限定性说明。

向量外积处理引擎 401 用于计算 N 个第一列向量和 N 个第一行向量的向量外积, 得到 N 个中间结果矩阵, 中间结果矩阵包括第三元素值及第三元素值的位置坐标, 位置坐标包括第一元素值的行坐标和第二元素值的列坐标。示例性的, 请参阅图 5B 所示, 向量外积处理引擎 401 计算第一列向量和第一行向量的向量外积。例如, 向量外积处理引擎 401 计算 A₀ 和 B₀ 的向量外积, 得到一个中间结果矩阵 C₀。同理, 向量外积处理引擎 401 计算 A₁ 和 B₁ 的向量外积, 得到一个中间结果矩阵 C₁。向量外积处理引擎 401 计算 A₂ 和 B₂ 的向量外积, 得到一个中间结果矩阵 C₂。向量外积处理引擎 401 计算 A₃ 和 B₃ 的向量外积, 得到一个中间结果矩阵 C₃。向量外积处理引擎 401 计算的中间结果矩阵以 C₀ 为例, 如下式 (11) 所示。

$$C_0 = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \otimes [b_0 \quad b_1 \quad b_2 \quad b_3] = \begin{bmatrix} a_0 b_0 & a_0 b_1 & a_0 b_2 & a_0 b_3 \\ a_1 b_0 & a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_0 & a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_0 & a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix}, \text{ 式 (11)}。$$

中间结果矩阵 C₀ 中包括第三元素值, 每个第三元素值具有对应的位置坐标。该位置坐标是第一元素值的行坐标和第二元素值的列坐标的结合并值。例如, 请参阅图 5C 所示, a₀ 的行坐标为 i₀, 而 b₀ 的列坐标为 j₀, 则第三元素值 a₀b₀ 的位置坐标为 (i₀, j₀)。同理, a₀ 的行坐标为 i₀, 而 b₁ 的列坐标为 j₁, 则第三元素值 a₀b₁ 的位置坐标为 (i₀, j₁)。a₁ 的行坐标为 i₁, 而 b₀ 的列坐标为 j₀, 则第三元素值 a₁b₀ 的位置坐标为 (i₁, j₀) 等。中间结果矩阵 C₀ 中的第三元素值的位置坐标不一一举例说明。

举例来说, a₀ 为 1, a₁ 为 2, a₂ 为 5, a₃ 为 6。且 a₀ 的行坐标为 “0”, 同理, a₁ 的行坐标为 “1”, a₂ 的行坐标为 “2”, a₃ 的行坐标为 “3”。b₀ 为 3, b₁ 为 5, b₂ 为 6, b₃ 为 7。b₀ 的列坐标为 “1”, b₁ 的列坐标为 “0”, b₂ 的列坐标为 “2”, b₃ 的列坐标为 “3”。第三元素值 a₀b₀ = 1 × 3 = 3, a₀b₀ 的位置坐标包括 a₀ 的行坐标和 b₀ 的列坐标, 即 a₀b₀ 的位置坐标为 (0, 3)。第三元素值 a₀b₁ = 1 × 5 = 5, a₀b₁ 的位置坐标包括 a₀ 的行坐标和 b₁ 的列坐标, 即 a₀b₁ 的位置坐标为 (0, 0)。第三元素值 a₁b₀ = 2 × 3 = 6, a₁b₀ 的位置坐标包括 a₁ 的行坐标和 b₀ 的列坐标, 即 a₁b₀ 的位置坐标为 (1, 3)。

请参阅图 5D 所示, 向量外积处理引擎 401 根据上述式 (7) 中的向量外积计算公式, 得到中间结果矩阵。向量外积处理引擎 401 计算的中间结果矩阵以 C_1 为例, 如下式 (12) 所示。

$$C_1 = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \otimes [d_0 \ d_1 \ d_2 \ d_3] = \begin{bmatrix} c_0 d_0 & c_0 d_1 & c_0 d_2 & c_0 d_3 \\ c_1 d_0 & c_1 d_1 & c_1 d_2 & c_1 d_3 \\ c_2 d_0 & c_2 d_1 & c_2 d_2 & c_2 d_3 \\ c_3 d_0 & c_3 d_1 & c_3 d_2 & c_3 d_3 \end{bmatrix}, \text{ 式 (12)}。$$

同理, 中间结果矩阵 C_1 中包括第三元素值, C_1 中每个第三元素值具有对应的位置坐标。

- 5 例如, 请参阅图 5D 所示, c_0 的行坐标为 k_0 , 而 d_0 的列坐标为 l_0 , 则第三元素值 $c_0 d_0$ 的位置坐标为 (k_0, l_0) 。同理, c_0 的行坐标为 k_0 , 而 d_1 的列坐标为 l_1 , 则第三元素值 $c_0 d_1$ 的位置坐标为 (k_0, l_1) 。 c_1 的行坐标为 k_1 , 而 d_0 的列坐标为 l_0 , 则第三元素值 $c_1 d_0$ 的位置坐标为 (k_1, l_0) 等。中间结果矩阵 C_1 中的第三元素值的位置坐标不一举例说明。

- 累加器 402, 用于根据第三元素值的位置坐标的索引, 将 N 个中间结果矩阵中位置坐标相同的第三元素值进行累加, 得到结果矩阵。请再次参阅图 5B 所示, 本示例中, 中间结果矩阵的数量以 4 个为例进行说明。中间结果矩阵 C_0, C_1, C_2 和 C_3 中每个中间结果矩阵中的第三元素值都具有位置坐标, 累加器 402 将位置坐标相同的第三元素值进行累加, 得到结果矩阵。例如, 请参阅图 5E 所示, 累加器 402 将 C_0, C_1, C_2 和 C_3 中位置坐标为 $(0, 0)$ 的 4 个第三元素值相加, 得到结果矩阵中的一个第四元素值。同理, 累加器 402 将 C_0, C_1, C_2 和 C_3 中位置坐标为 $(1, 1)$ 的 4 个第三元素值相加, 得到结果矩阵中的一个第四元素值等。本示例中, 矩阵位置坐标的具体数值仅是为了方便说明而举的例子, 并非限定。

可选地, 累加器 402 根据第三元素值的位置坐标的索引, 将 N 个中间结果矩阵中位置坐标相同的第三元素值进行累加的实现方式至少包括以下两种实现方式。

- 第一种可以实现的方式中, 向量外积处理引擎 401 在对第一行向量和第一列向量进行向量外积计算时, 会按照一定顺序产生中间结果矩阵。例如, 以上述图 5B 中示出的第一行向量和第一列向量计算向量外积为例。向量外积处理引擎 401 计算 A_0 和 B_0 的向量外积, 得到一个中间结果矩阵 C_0 。然后, 向量外积处理引擎 401 计算 A_1 和 B_1 的向量外积, 得到一个中间结果矩阵 C_1 。再后, 向量外积处理引擎 401 计算 A_2 和 B_2 的向量外积, 得到一个中间结果矩阵 C_2 。最后, 向量外积处理引擎 401 计算 A_3 和 B_3 的向量外积, 得到一个中间结果矩阵 C_3 。例如, 4 个中间结果矩阵的顺序为 C_0, C_1, C_2, C_3 。累加器 402 按照 4 个中间结果矩阵的产生顺序, 从向量外积处理引擎 401 接收 4 个中间结果矩阵。

- 请参阅图 6 所示, 4 个中间结果矩阵至少包括第一中间结果矩阵 (如 C_0) 和第二中间结果矩阵 (如 C_1)。为了区分第一中间结果矩阵中的位置坐标和第二中间结果矩阵中的位置坐标, 将第一中间结果矩阵中第三元素值的位置坐标称为“第一位置坐标”, 将第二结果矩阵中第三元素值的位置坐标称为“第二位置坐标”。

- 示例性的, 在第一缓存器 403 中, 通过行坐标标识和纵坐标标识来将第一缓存器 403 划分为多个存储位置。例如, 在第一缓存器 403 中, 通过 p 个行 q 个列将第一缓存器 403 的存储空间划分为 $q \times p$ 个存储位置。累加器 402 接收到 C_0 后, 将 C_0 中第三元素值按照第一位置坐标写入第一缓存器 403 中的对应位置。例如, C_0 中第三元素值 $a_0 b_0$ 的位置坐标 (i_0, j_0) , 如 (i_0, j_0) 为 $(1, 1)$, 那么累加器 402 按照位置坐标 (i_0, j_0) 将 $a_0 b_0$ 写入第一缓存器 403 中的第 1 行第 1 列的位置。同理, C_0 中第三元素值 $a_0 b_1$ 的位置坐标 (i_0, j_1) , 如 (i_0, j_1) 为 $(1, 2)$, 累加器 402 按照位置坐标 (i_0, j_1) 将 $a_0 b_1$ 写入第一缓存器 403 中的第 1 行第 2 列的位置。

C_0 中第三元素值 a_0b_2 的位置坐标 (i_0, j_2) , 如 (i_0, j_2) 为 $(1, 3)$, 累加器 402 按照位置坐标 (i_0, j_2) 将 a_0b_2 写入第一缓存器 403 中的第 1 行第 3 列的位置。 C_0 中第三元素值 a_0b_3 的位置坐标 (i_0, j_3) , 如 (i_0, j_3) 为 $(1, 4)$, 累加器 402 按照位置坐标 (i_0, j_3) 将 a_0b_3 写入第一缓存器 403 中的第 1 行第 4 列的位置。累加器 402 将 C_0 中的其他第三元素值写入第一缓存器 403 的过程不一一赘述说明。最终的结果是, 累加器 402 将 C_0 中的所有第三元素值按照每个第三元素值对应的第一位置坐标将第三元素值写入第一缓存器 403。

然后, 当累加器 402 接收到 C_1 时, 累加器 402 按照 C_1 中第三元素值的第二位置坐标查找第一缓存器 403 中第二位置坐标的对应位置是否有缓存值, 若第二位置坐标的对应位置没有缓存值, 累加器 402 将第三元素值写入第一缓存器 403 中的第二位置坐标的对应位置。若第二位置坐标的对应位置有缓存值, 则累加器 402 从第一缓存器 403 中读取该缓存值, 将该缓存值与该 C_1 中第三元素值进行累加。举例说明, C_1 中的第三元素值 c_0d_0 的位置坐标 (k_0, l_0) , 如 (k_0, l_0) 为 $(1, 0)$, 累加器 402 按照位置坐标 (k_0, l_0) 查询第一缓存器 403, 如果在第一缓存器 403 中的第 1 行第 0 列的位置没有缓存值, 那么累加器 402 将 c_0d_0 写入到第一缓存器 403 中的第 1 行第 0 列的位置。 C_1 中的第三元素值 c_0d_1 的位置坐标 (k_0, l_1) , 如 (k_0, l_1) 为 $(1, 1)$, 累加器 402 查询第一缓存器 403 中第 1 行第 1 列的位置有缓存值 a_0b_0 , 累加器 402 读取 a_0b_0 , 并将 c_0d_1 和 a_0b_0 进行相加, 然后将相加后的结果 $(c_0d_1 + a_0b_0)$ 按照 c_0d_1 的位置坐标再写入第一缓存器 403 的对应位置 (第 1 行第 1 列)。同理, C_1 中的第三元素值 c_0d_2 的位置坐标 (k_0, l_2) , 如 (k_0, l_2) 为 $(1, 2)$, 累加器 402 查询第一缓存器 403 中第 1 行第 2 列的位置有缓存值 a_0b_1 , 累加器 402 从第一缓存器 403 读取该缓存值 a_0b_1 , 并将 c_0d_2 和 a_0b_1 相加, 然后将相加后的累加值 $(c_0d_2 + a_0b_1)$ 按照 c_0d_2 的位置坐标再写入第一缓存器 403 的对应位置 (第 1 行第 2 列)。本示例中, 对于 C_1 中的第三元素值不一一举例说明。最终的结果是, 累加器 402 将 C_1 和 C_0 中位置坐标相同的第三元素值相加, 相加后的结果按照该位置坐标写入第一缓存器 403 中的对应位置。同理, 当累加器 402 接收到向量外积处理引擎 401 传输的中间结果矩阵 C_2 时, 累加器 402 按照 C_2 中的每个第三元素值的位置坐标从第一缓存器 403 的对应位置读取缓存值 (C_0 和 C_1 中位置坐标相同的第三元素值的累加值), 累加器 402 将 C_0 、 C_1 和 C_2 中位置坐标相同的第三元素值的累加, 然后将累加后的结果再写入第一缓存器 403 中位置坐标的对应位置。应理解, 累加器 402 对 C_2 和 C_3 的处理过程, 与对 C_1 的处理过程类似, 此处不一一赘述。累加器 402 最终的处理结果是, 将 C_0 、 C_1 、 C_2 和 C_3 中位置坐标相同的第三元素值进行累加, 得到第五元素值, 结果矩阵包括多个第五元素值。最后, 第一缓存器 403 输出该结果矩阵。

需要说明的是, 第五元素值是 C_0 、 C_1 、 C_2 和 C_3 中至少一个中间结果矩阵中的第三元素值的累加值。举例来说, 请再次参阅图 6 所示, C_0 中的 a_0b_0 的位置坐标是 $(1, 0)$, 而在其他三个中间结果矩阵 (C_1 、 C_2 和 C_3) 中所有第三元素值的位置坐标都没有 $(1, 0)$, 那么在第一缓存器 403 第 1 行第 0 列的位置上写入的值就是 a_0b_0 , 该 a_0b_0 不会和其他中间结果矩阵中的第三元素值进行累加。同理, C_0 中的 a_0b_1 和 C_1 中的 c_0d_1 的位置坐标都是 $(1, 1)$, 而在其他两个中间结果矩阵 (C_2 和 C_3) 中的第三元素值的位置坐标都没有 $(1, 1)$, 那么在第一缓存器 403 第 1 行第 1 列的位置上写入的值就是 a_0b_1 和 c_0d_1 的累加值。综上, 应理解, 一个第五元素值可能是一个第三元素值, 也可能是两个第三元素值的累加值, 也可能是三个第三元素值的累加值, 也可能是四个第三元素值的累加值等。在实际运算中, 结果矩阵中某一个第五元素值是由几个第三元素值累加得到的, 由相同的位置坐标下对应的第三元素值的个数确定, 具体

的并不限定。

进一步的，第五元素值是第三元素值的累加值，那么，第五元素值可能为零元素值。本示例中，结果矩阵为非压缩格式的矩阵。可选地，为了节省传输资源或便于下一步计算操作，矩阵计算装置可以将结果矩阵进行压缩，以输出压缩格式的矩阵。请参阅图 7 所示，图 7 为矩阵计算装置的另一个结构示意图。矩阵计算装置和矩阵压缩单元 404，矩阵压缩单元 404 与第一缓存器 403 连接，第一缓存器 403 输出结果矩阵，矩阵压缩单元 404 接收结果矩阵，按照第一缓存器 403 中的行坐标标识和列坐标标识将结果矩阵转换成压缩格式的矩阵。结果矩阵可以为 COO 格式，CSR 或 CSC 格式等，具体的并不限定。

应理解，累加器 402 在进行累加计算的过程中，第一缓存器 403 用于存储第三元素值及将第三元素值进行累加之后的结果。该第一缓存器 403 的存储空间需要大于或者等于某一个数值。例如，若每个中间结果矩阵的维度为 $M \times P$ ，也即每个中间结果矩阵包括 $M \times P$ 个第三元素值。第一缓存器 403 包括的缓存位置的数量大于或者等于 $M \times P$ 。那么该第一缓存器 403 的存储空间需要至少能够存储 $M \times P$ 个数值。

在第一种可能实现的方式中，矩阵计算装置可以根据不同的应用场景输出非压缩格式的矩阵或者输出压缩格式的矩阵，增加了矩阵计算装置的适用场景。

在第二种可能实现的方式中，请参阅图 8 所示，累加器 402 将 N 个中间结果矩阵中第三元素值按照第三元素值的位置坐标进行排序，然后将排序后的 N 个中间结果矩阵中的位置坐标进行比对，将位置坐标相同的第三元素值相加，并删除零元素值的位置坐标，得到压缩格式的结果矩阵。例如，当累加器 402 接收到 C_0 时，累加器 402 先将 C_0 中的第三元素值按照第三元素值的位置坐标进行排序后写入第一缓存器 403 内，累加器 402 可以按照第三元素值的行坐标进行排序，可选地，累加器 402 可以按照第三元素值的列坐标进行排序。本示例中，以累加器 402 按照第三元素值的行坐标排序为例进行说明。累加器 402 接收到 C_1 ，将 C_1 中的第三元素值按照第三元素值的行坐标进行排序。累加器 402 按照 C_1 中第三元素值的位置坐标的顺序将 C_1 中第三元素值的位置坐标与 C_0 中的第三元素值的位置坐标进行比对。例如，累加器 402 首先比对行坐标，如比对 k_0 与 i_0 ，若 k_0 与 i_0 一致，则继续比对列坐标。若 k_0 与 i_0 不一致，累加器 402 继续将 k_0 与 i_1 比对。累加器 402 比对完行坐标后，按照位置坐标的顺序比对标坐标 l_0 与 j_0 ， l_0 与 j_1 等，当位置坐标 (k_0, l_0) 与位置坐标 (i_0, j_1) 一致时，累加器 402 将位置坐标 (k_0, l_0) 对应的第三元素值 $c_0 d_0$ 和位置坐标 (i_0, j_1) 对应的第三元素值 $a_0 b_0$ 相加，然后将相加之后的结果写入第一缓存器 403。

同理，当累加器 402 接收到 C_2 时，累加器 402 将 C_2 中的第三元素值按照位置坐标进行排序，累加器 402 读取第一缓存器 403 中的位置坐标，然后将 C_2 中的第三元素值的位置坐标与第一缓存器 403 中的位置坐标进行比对，将位置坐标相同的第三元素值和缓存值相加，得到累加值，然后将累加值存储到第一缓存器 403。累加器 402 对 C_3 中第三元素值的计算与对 C_2 中的第三元素值的计算相似，此处不赘述举例说明。最终的结果是，累加器 402 将中间结果矩阵 C_3 ， C_2 ， C_1 和 C_0 中位置坐标相同的第三元素值累加，得到第五元素值，并且累加器 402 删除多个第五元素值中的零元素值及零元素值的位置坐标。第一缓存器 403 输出结果矩阵，该结果矩阵包含第五元素值及第五元素值的位置坐标。且结果矩阵中包括的第五元素值为非零元素值，由此，第一缓存器 403 输出的结果矩阵为压缩格式的矩阵。

在第二种实现方式中，可以直接得到压缩格式的矩阵，矩阵计算装置输出的压缩格式的矩阵可以应用于某些后续计算需要压缩格式的矩阵的应用场景。且由于矩阵计算装置输出的

是压缩格式的矩阵，可以减少后续传输矩阵的传输资源。而且在第二种实现方式中，由于第一缓存器 403 中缓存的第三元素值是按位置坐标的顺序进行缓存，较少的缓存空间就可以实现本示例中的实现方式，从而节省第一缓存器 403 的缓存空间。

可选地，本示例中，格式转换单元 405，还用于获取第五矩阵和第六矩阵，对非压缩格式的第五矩阵和第六矩阵进行格式转换，得到压缩格式的第一矩阵和第二矩阵。并将第一矩阵和第二矩阵输出至第二缓存器 406，其中，第五矩阵和第六矩阵中至少有一个矩阵为非压缩格式的矩阵。向量外积处理引擎 401 从第二缓存器 406 获取压缩格式的第一矩阵和第二矩阵。本示例中，矩阵计算装置可以接收非压缩格式的矩阵，然后将非压缩格式的矩阵转换成压缩格式的矩阵，从而使得矩阵计算装置可以支持多种格式的矩阵计算。本示例中，第五矩阵和第六矩阵可以包括如下几种情况。

第一种情况：请参阅图 9 所示，第五矩阵和第六矩阵均为非压缩格式的矩阵。这种情况中，格式转换单元 405 将第五矩阵为压缩格式的第一矩阵，将第六矩阵转换为压缩格式的第二矩阵。可选地，格式转换单元 405 可以将第五矩阵和第六矩阵转换成 COO 格式的压缩矩阵。可选地，由于 CSC 压缩格式中保留了元素值的行坐标，而 CSR 压缩格式中保留了元素值的列坐标，格式转换单元可以将非压缩格式的第五矩阵转换成 CSC 压缩格式的第一矩阵，而将非压缩格式的第六矩阵转换成 CSR 压缩格式。进一步的，格式转换单元 405 将压缩格式的第一矩阵转换为 N 个第一列向量，将压缩格式的第二矩阵转换为 N 个第一行向量，并将 N 个第一列向量和 N 个第一行向量写入第一缓存器 406 中，向量外积处理引擎 401 从第二缓存器 406 中获取第一行向量和第二行向量。

第二种情况：第五矩阵和第六矩阵中有一个矩阵为非压缩格式的矩阵，而另一个矩阵为压缩格式的矩阵。第五矩阵为非压缩格式的矩阵，第六矩阵为压缩格式的矩阵为例进行说明。若第六矩阵为 CSC 或 CSR 格式的压缩矩阵。格式转换单元将第五矩阵和第六矩阵均转换成 COO 格式的压缩矩阵。

可选地，格式转换单元 405，还用于将压缩格式的矩阵转换成目标压缩格式的矩阵。例如，第一矩阵和第二矩阵均为 CSC 格式或 CSR 格式时，格式转换单元将第一矩阵和第二矩阵均转换成 COO 格式。

本示例中，矩阵计算装置通过格式转换单元可以将非压缩格式的矩阵转换为压缩格式的矩阵，从而使矩阵计算装置既可以支持压缩格式的矩阵计算，又可以支持非压缩格式的矩阵计算。可选地，格式转换单元可以将非目标压缩格式的矩阵转换成目标压缩格式（如 COO 格式）的矩阵。本示例中，矩阵计算装置可以将其他压缩格式的矩阵转换成目标压缩格式的矩阵，对目标压缩格式的矩阵进行矩阵计算，本申请中提供的矩阵计算装置可以支持各种格式的矩阵计算。

可选地，为了提高矩阵计算单元的适用性，可基于低精度的矩阵计算装置实现高精度的矩阵计算。第一列向量和第一行向量中包括的元素值的精度为第一精度。格式转换单元 405 将第一列向量分解成 X 个第二列向量，并将第一行向量分解成 X 个第二行向量，其中，第二列向量和第二行向量包含的第二精度的元素值，第一精度高于第二精度。然后，向量外积处理引擎 401 计算 X 个第二列向量和 X 个第二行向量的向量外积，得到 X^2 个第四矩阵，第四矩阵包括第四元素值及第四元素值的位置坐标。最后，累加器 402 根据第四元素值的位置坐标的索引，将 X^2 个第四矩阵中位置坐标相同的第四元素值累加，得到中间结果矩阵，中间结果

矩阵中的第三元素值的精度为第一精度。

下面对本示例中将第一精度的向量分解成第二精度的向量进行举例说明。格式转换单元 405 将压缩格式的第一矩阵转换成 N 个第一列向量，将压缩格式的第二矩阵转换成 N 个第一行向量之后，还可以进一步的按照元素值的精度将一个第一列向量分解成 X 个第二列向量，并将第一行向量分解成 X 个第二行向量。其中，第一列向量和第一行向量中包括的元素值的精度为第一精度。第二列向量和第二行向量均包含第二精度的元素值。为了方便说明，将包含第二精度的元素值的向量称为“第二精度的向量”，将包含第一精度的元素值的向量称为“第一精度的向量”。X 为大于或者等于 2 的整数。第一精度和第二精度可以为整型精度，或者，第一精度和第二精度也可以是浮点数精度，具体的并不限定。具体的，格式转换单元 405 将第一精度的向量中的每一个元素值拆分成多个第二精度的数值，将第一精度的向量拆解成 X 个第二精度的向量。下面按照第一精度的向量和第二精度的向量是整型，及第一精度的向量和第二精度的向量是浮点型向量这种情况进行说明。

第一种情况，第一精度的向量和第二精度的向量是整型。第一精度高于第二精度，例如，第一精度为 int32，第二精度可以为 int2、int4、int8 或 int16。或者，第一精度为 int16，第二精度可以为 int4 或 int8。或者，第一精度为 int8，第二精度为 int2 或 int4。或者，第一精度为 int4，第二精度为 int2 等。

格式转换单元 405 将一个高精度的整型数值拆分成多个低精度的整型数值，将高精度的整型数值按照从高位到低位的顺序进行拆分。例如，请参阅图 10 所示，以第一精度为 int32，第二精度为 int16 为例进行说明。int32 包括 32bit 值域，将一个 int32 中的 32 位从高位到低位的顺序拆分成 2 个 16bit，即将 int32 拆分成 2 个 int16。或者，将一个 int32 中的 32 位从高位到低位的顺序拆分成 4 个 8 bit，即将 int32 拆分成 4 个 int8。同理，int16 包括 16bit 值域，将一个 int16 中的 16 位从高位到低位的顺序拆分成 2 个 8bit，即将 int16 拆分成 2 个 int8。

第二种情况，第一精度的向量和第二精度的向量是浮点型。例如，第一精度为 FP32、FP64，第二精度为 FP16。或者，第一精度 FP64，第二精度浮点可以为 FP32，也可以为 FP16。下面以第一精度浮点数为 FP32，第二精度浮点数为 FP16 的情况为例进行说明。

一、将一个 FP32 进行分解，得到三个 FP16。

目前，标准格式的 FP32 的组成如上表 1 所示，FP32 包括 1bit（比特）符号，8bit 阶码和 23bit 尾数。另外，还有省略的 1bit 整数，省略的整数为 1。对于一个标准格式的 FP32，其整数加上尾数，共 24bit。标准格式的 FP16 的组成包括 1bit 符号，5bit 阶码和 10bit 尾数，另外，还有省略的 1bit 整数，省略的整数为 1。对于一个标准格式的 FP16 整数加上尾数，共 11bit。如果要将一个标准格式的 FP32 分解，得到标准格式的 FP16，需要 3 个标准格式的 FP16。

将标准格式的 FP32 的整数和尾数可以分为三部分，第一部分为整数和前 10bit 的尾数，第二部分为第 11bit 到第 21bit 的尾数，第三部分为第 22bit 和第 23bit 的尾数。将这三部分分别由一个标准格式的 FP16 表示。此处需要说明的是，对于上述第三部分第 22bit 和第 23bit 的尾数由标准格式的 FP16 表示时，可以先在第 23bit 的尾数后补 9 个 0，即将第 22bit 和第 23bit 的尾数，以及补的 0，由一个标准格式的 FP16 表示。

另外，FP16 的阶码范围为-15 到 15，即可以表示小数点左移 15 位到右移 15 位。使用标准格式的 FP16 表示上述 FP32 的第一部分时，固定阶码位移值为 0；使用标准格式的 FP16

表示上述 FP32 的第二部分时, 固定阶码位移值为-11; 使用标准格式的 FP16 表示上述 FP32 的第三部分时, 固定阶码位移值为-22。可见, 在表示第三部分时, 仅对应的固定阶码位移值就已经超出了 FP16 的阶码范围。因此, 可以对每个标准格式的 FP16 的阶码提取出对应的固定阶码位移值。

5 那么, 对于一个标准格式的 FP32, 可以表示为:

$A_1 = 2^{EA_1} (a_0 + 2^{-S_1} a_1 + 2^{-2S_1} a_2)$, 其中, A_1 为标准格式的 FP32, EA_1 为 A_1 的阶码, a_0 、 a_1 和 a_2 为分解得到的三个标准格式的 FP16, S_1 为最小的固定阶码位移值, 对于该标准格式的 FP16 来说, $S_1 = 11$ 。

10 除此之外, 对于每个标准格式的 FP16 的阶码可以提取出一个公共阶码位移值。那么, 同样对于一个标准格式的 FP32, 可以表示为:

$A_1 = 2^{EA_1 - S_1} (a_0' + a_1' + a_2')$, 其中, a_0' 、 a_1' 和 a_2' 为分解得到的三个标准格式的 FP16。在上述两种表示方法中, 分解得到的 FP16 有如下关系: $a_0 = 2^{-S_1} a_0'$, $a_1 = a_1'$, $a_2 = 2^{S_1} a_2'$ 。

二、将一个 FP32 分解, 得到两个 FP16。

15 为了减少分解得到的 FP16 的个数, 对于目前标准格式的 FP16 可以进行调整, 将其尾数调整为 13bit, 符号和阶码的 bit 数不变, 可以称调整后的 FP16 为非标准格式的 FP16。则该非标准格式的 FP16 的整数加上尾数, 共 14bit。那么, 如果要将一个标准格式的 FP32 的尾数使用非标准格式的 FP16 表示, 则只需要 2 个非标准格式的 FP16。

将标准格式的 FP32 的整数和尾数分为两部分, 第一部分为整数和前 13bit 的尾数, 第二部分为第 14bit 到第 23bit。将这两部分分别由非标准格式的 FP16 表示。

20 此处还需说明的是, 对于上述第二部分由非标准 FP16 表示时, 可以先在第 23bit 的尾数后补 4 个 0, 即将第 14bit 到第 23bit 的尾数, 以及补的 0, 由一个非标准格式的 FP16 表示。与上述情况一相同, 此处, 也可以对每个标准格式的 FP16 的阶码提取出对应的固定阶码位移值。

那么, 同样对于一个标准格式的 FP32, 可以表示为:

25 $A_2 = 2^{EA_2} (a_3 + 2^{-S_2} a_4)$, 其中, A_2 为标准格式的 FP32, EA_2 为 A_2 的阶码, a_3 和 a_4 为分解得到的两个非标准格式的 FP16, S_2 为固定阶码位移值, 对于该非标准格式的 FP16 来说, $S_2 = 14$ 。

除此之外, 对于每个标准格式的 FP16 的阶码可以提取出一个公共阶码位移值。那么, 同样对于一个标准格式的 FP32, 可以表示为:

30 $A_2 = 2^{EA_2 - S_2} (a_3' + a_4')$, 其中, a_3' 和 a_4' 为分解得到的两个非标准格式的 FP16。在上述两种表示方法中, 分解得到的 FP16 有如下关系: $a_3 = 2^{-S_2} a_3'$, $a_4 = a_4'$ 。

35 当然, 针对第一精度浮点数为 FP64, 第二精度浮点数为 FP32 的情况, 对于 FP64 分解得到多个 FP32 可以有如下情况: 将一个 FP64 浮点数进行分解, 得到三个 FP32 浮点数; 或者, 将一个 FP64 浮点数分解, 得到两个 FP32 浮点数。可选地, 对于第一精度浮点数为 FP64, 第二精度浮点数为 FP16 的情况, 对于 FP64 分解得到多个 FP16 可以有如下情况: 将一个 FP64 浮点数进行分解, 得到五个 FP16 浮点数; 或者, 将一个 FP64 浮点数分解, 得到 4 个 FP16 浮点数。拆分原理与上述描述的第一精度浮点数为 FP32, 第二精度浮点数为 FP16 的情况类似, 此处不一一赘述。

40 示例性的, 为了方便说明, 以将第一行向量拆分成两个第二行向量, 将第一列向量拆分成两个第二列向量为例进行说明。例如, 第一行向量 A_0 为一个精度为 FP32 的列向量

$[a_0, a_1, a_2, a_3]^T$ ，如上述将第一精度的数值拆分成两个第二精度的数值的方法，格式转换单元 405 将精度为 FP32 的浮点数 a_0 拆分成 2 个 FP16 的浮点数（如 a_{0M} 和 a_{0L} ）。同理，将 a_1 拆分成 a_{1M} 和 a_{1L} 。将 a_2 拆分成 a_{2M} 和 a_{2L} 。将 a_3 拆分成 a_{3M} 和 a_{3L} 。即请参阅 11 所示，格式转换单元 405 将精度为 FP32 的列向量 $[a_0 \ a_1 \ a_2 \ a_3]^T$ 拆分为两个精度为 FP16 的列向量

5 $[a_{0M} \ a_{1M} \ a_{2M} \ a_{3M}]^T$ （记做“第二列向量 1”）与 $[a_{0L} \ a_{1L} \ a_{2L} \ a_{3L}]^T$ （记做“第二列向量 2”）。同理， B_0 是一个精度 FP32 的行向量 $[b_0 \ b_1 \ b_2 \ b_3]$ ，将 $[b_0 \ b_1 \ b_2 \ b_3]$ 也拆分为两个 FP16 的行向量 $[b_{0M} \ b_{1M} \ b_{2M} \ b_{3M}]$ （记做“第二行向量 1”）与 $[b_{0L} \ b_{1L} \ b_{2L} \ b_{3L}]$ （记做“第二行向量 2”）。需要说明的是，第一列向量和第二列向量中每个元素值的行坐标相同，即 a_{0M} 和 a_{0L} 的行坐标为 i_0 ， a_{1M} 和 a_{1L} 的行坐标为 i_1 ， a_{2M} 和 a_{2L} 的行坐标为 i_2 。 a_{3M} 和 a_{3L} 的行

10 坐标为 i_3 。第一行向量和第二行向量中每个元素值的列坐标相同，即 b_{0M} 和 b_{0L} 的列坐标为 j_0 ， b_{1M} 和 b_{1L} 的列坐标为 j_1 ， b_{2M} 和 b_{2L} 的列坐标为 j_2 ， b_{3M} 和 b_{3L} 的列坐标为 j_3 。

进一步的，向量外积处理引擎 401 计算 X 个第二列向量和 X 个第二行向量的向量外积，得到 X^2 个第四矩阵，第四矩阵包括第四元素值及第四元素值的位置坐标，第四元素值的位置坐标包括第一列向量中元素值的行坐标和第一行向量中元素值的列坐标，第四元素值的精度为第一精度（如 FP32）。本示例中，请参阅图 12 所示，X 以 2 为例进行说明。向量外积处理引擎 401 计算 $[a_{0M} \ a_{1M} \ a_{2M} \ a_{3M}]^T$ 和 $[a_{0L} \ a_{1L} \ a_{2L} \ a_{3L}]^T$ 这两个第二列向量与 $[b_{0M} \ b_{1M} \ b_{2M} \ b_{3M}]$ 和 $[b_{0L} \ b_{1L} \ b_{2L} \ b_{3L}]$ 这两个第二行向量的向量外积。即向量外积处理引擎 401 计算 $[a_{0M} \ a_{1M} \ a_{2M} \ a_{3M}]^T$ 和 $[b_{0M} \ b_{1M} \ b_{2M} \ b_{3M}]$ 的向量外积，得到第四矩阵 1；计算 $[a_{0M} \ a_{1M} \ a_{2M} \ a_{3M}]^T$ 和 $[b_{0L} \ b_{1L} \ b_{2L} \ b_{3L}]$ 的向量外积，得到第四矩阵 2；计算

20 $[a_{0L} \ a_{1L} \ a_{2L} \ a_{3L}]^T$ 和 $[b_{0M} \ b_{1M} \ b_{2M} \ b_{3M}]$ 的向量外积，得到第四矩阵 3； $[a_{0L} \ a_{1L} \ a_{2L} \ a_{3L}]^T$ 和 $[b_{0L} \ b_{1L} \ b_{2L} \ b_{3L}]$ 的向量外积，得到第四矩阵 4。向量外积处理引擎 401 通过向量外积计算得到 4 个第四矩阵。累加器 402 获取这 4 个第四矩阵，并根据第四元素值的位置坐标的索引，将 4 个第四矩阵中位置坐标相同的第四元素值累加，得到一个中间结果矩阵，中间结果矩阵中的第三元素值的精度为第一精度。例如，第四矩阵 1 中的第四元素值 $a_{0M}b_{0L}$ 、第四矩阵 2 中的第四元素值 $a_{0M}b_{1M}$ 、第四矩阵 3 中的第四元素值 $a_{1L}b_{0L}$ ，第四矩阵 4 中的第四元素值 $a_{1L}b_{1M}$ 的位置坐标相同，累加器 402 将 $a_{0M}b_{0L}$ 、 $a_{0M}b_{1M}$ 、 $a_{1L}b_{0L}$ 和 $a_{1L}b_{1M}$ 这 4 个第四元素值累加，得到的累加值为中间结果矩阵 C_0 中的一个第三元素值。本示例中，将 X^2 个第四矩阵中位置坐标相同的第四元素值累加的具体方法与参阅图 6 和图 8 对应的示例中累加器 402 进行累加的方法相似，具体请参阅上述对图 5E，图 6 和图 8 对应的示例中累加器 402 执行的方法，此处不赘述。本示例中，在计算两个向量的向量外积时，可以进一步的将第一精度（高精度）的向量拆分成多个第二精度（低精度）的向量，然后对低精度的向量进行向量外积计算，从而使得第一精度的向量的外积可由多个第二精度的向量的外积结果做累积而得，且精度不会损失。

25 30

本示例中，若第一行向量和第一列向量中的元素值的精度较大，矩阵计算装置可以将第一行向量和第一列向量均拆解为多个低精度的向量。矩阵计算装置对低精度的第二列向量和第二行向量进行向量外积计算，以得到结果矩阵，既能够对压缩格式的矩阵进行计算，而且能够基于低精度的矩阵计算装置实现高精度的矩阵计算，提高了矩阵计算装置的适用性。并且，在矩阵计算过程中，基于矩阵计算装置的上层软件应用（如 AI 和 HPC）不感知矩阵计算的具体过程，从而可以极大地降低软件适配的成本。

35

40 基于本申请中提供的矩阵计算装置，能在多个矩阵计算场景得到极大的收益，如将矩阵

计算装置使用在AI训练与推理场景中，可完整支持压缩格式的矩阵和非压缩格式的矩阵计算，针对AI计算中的权重与特征数据的稀疏特性平均达50%以上（即50%以上为压缩格式的矩阵），本申请中的矩阵计算格式可以直接对压缩格式的矩阵进行计算，不需要对压缩格式的矩阵进行解压缩超出，这样计算效能可提高4倍以上。另针对科学计算等HPC场景，不论是高算力要求的非压缩格式的矩阵计算或内存带宽受限的矩阵计算场景中，使用本申请中的矩阵计算装置能直接从内存存取压缩格式的矩阵，提升计算收益。矩阵计算装置支持全精度数值计算，也能有效的覆盖各种不同精度需求的计算，如AI训练场景常需要FP32和FP16等浮点数计算，又如一些需要FP64的AI训练场景与HPC科学计算等场景，矩阵计算装置都能完整支持。并且，矩阵计算装置中的MAC也能支持中低精度如INT1、INT2、INT4、INT8等整数格式计算，针对AI推理的计算场景，能提升算力与并减少推理计算时间，也能提供混合各种不同精度于推理计算的场景中，极大的增强了矩阵计算装置的适用性。

上面对一种矩阵计算装置的实施例进行了说明，下面对矩阵计算装置所执行的方法进行说明。请参阅图 13 所示，本申请实施例提供了一种矩阵计算的方法，该方法的执行主体可以是图 2 所示的计算设备，可选地，该方法的执行主体是图 3 中所示出的矩阵计算装置，可选的，该方法的执行主体可以是图 4A 所示的矩阵计算装置，可选地，该方法的执行主体可以是图 7 所示的矩阵计算装置。

步骤 1301、获取第一计算指令，第一计算指令包括 N 个第一列向量和 N 个第一行向量。N 个第一列向量是由压缩格式的第一矩阵转换得到的，N 个第一行向量是由压缩格式的第二矩阵转换得到的，N 为大于或者等于 1 的整数。

步骤 1302、计算 N 个第一列向量和 N 个第一行向量的向量外积，得到 N 个中间结果矩阵，其中，第一列向量包括第一元素值及第一元素值的行坐标，第一行向量包括第二元素值及第二元素值的列坐标，中间结果矩阵包括第三元素值及第三元素值的位置坐标，位置坐标包括第一元素值的行坐标和第二元素值的列坐标。

本步骤请参阅上述矩阵计算装置实施例中，请参阅图 5B，图 5C 和图 5D 对应的示例中，向量外积处理引擎 401 所执行的功能的具体说明，此处不赘述。

步骤 1303、根据第三元素值的位置坐标的索引，将 N 个中间结果矩阵中位置坐标相同的第三元素值进行累加，得到结果矩阵。

N 个中间结果矩阵至少包括第一中间结果矩阵和第二中间结果矩阵，第一中间结果矩阵中第三元素值的位置坐标为第一位置坐标，第二中间结果矩阵中第三元素值的位置坐标为第二位置坐标。

在第一种可能实现的方式中，矩阵计算装置按照 N 个中间结果矩阵的产生顺序，将第一中间结果矩阵中第三元素值按照第一位置坐标写入缓存器中的对应位置。然后根据第二中间结果矩阵中第三元素值的第二位置坐标，读取第二位置坐标在缓存器中对应位置的缓存值，将第二中间结果矩阵中第三元素值和缓存值进行累加，得到非压缩格式的结果矩阵。可选地，矩阵计算装置压缩非压缩格式的结果矩阵，得到压缩格式的结果矩阵。

在第一种可能的实现方式中，请参阅上述矩阵计算装置实施例中，图 5E 和图 6 对应的示例中，累加器 402 所执行的功能的具体说明，此处不赘述。

在第二种可能的实现方式中，矩阵计算装置将 N 个中间结果矩阵中第三元素值按照第三

元素值的位置坐标进行排序。矩阵计算装置将排序后的 N 个中间结果矩阵中的位置坐标进行比对，将位置坐标相同的第三元素值相加，并删除零元素值的位置坐标，得到压缩格式的结果矩阵。

在第二种可能的实现方式中，请参阅上述矩阵计算装置实施例中，图 5E 和图 8 对应的示例中，累加器 402 所执行的功能的具体说明，此处不赘述。

本申请实施例中，矩阵计算装置能够直接对压缩格式的矩阵进行计算，不需要向传统方法那样将压缩格式的矩阵进行解压缩，对解压缩后的矩阵进行矩阵计算，本申请实施例中的矩阵计算装置能够提升压缩格式的矩阵的计算效能。

10 可选地，请参阅图 14 所示，为了能够支持支持多种格式的矩阵计算，矩阵计算装置能够将非压缩格式的矩阵转换成压缩格式的矩阵，从而可以对非压缩格式的矩阵进行计算，对非压缩格式的矩阵进行格式转换如下步骤 1401 和步骤 1402 所示。

步骤 1401、获取第三计算指令，第三计算指令包括第五矩阵和第六矩阵，其中，第五矩阵和第六矩阵中至少有一个矩阵为非压缩格式的矩阵。

15 步骤 1402、对第五矩阵进行格式转换，得到压缩格式的第一矩阵，对第六矩阵进行格式转换，得到第二矩阵。

步骤 1401 和步骤 1402 请参阅上述矩阵计算装置的实施例中，图 9 对应的示例中，格式转换单元 405 所执行的功能的具体说明，此处不赘述。

20 本申请实施例中，矩阵计算装置通过格式转换单元可以将非压缩格式的矩阵转换为压缩格式的矩阵，从而使矩阵计算装置既可以支持压缩格式的矩阵计算，又可以支持非压缩格式的矩阵计算。可选地，格式转换单元可以将非目标压缩格式的矩阵转换成目标压缩格式（如 COO 格式）的矩阵。本示例中，矩阵计算装置可以将其他压缩格式的矩阵转换成目标压缩格式的矩阵，对目标压缩格式的矩阵进行矩阵计算，本申请中提供的矩阵计算装置可以支持各种格式的矩阵计算。

25 步骤 1403、获取第二计算指令，第二计算指令包括第一矩阵和第二矩阵。

步骤 1404、将第一矩阵转换成 N 个第一列向量，将第二矩阵转换成 N 个第一行向量。

步骤 1403 和步骤 1404 请参阅上述矩阵计算装置的实施例中，图 5A 对应的示例中，格式转换单元 405 所执行的功能的说明，此处不赘述。

30 可选地，为了提高矩阵计算单元的适用性，可基于低精度的矩阵计算装置实现高精度的矩阵计算。将第一精度的向量拆分成多个第二精度的向量，然后对第二精度的向量进行向量外积计算。请参阅下述步骤 1405 至步骤 1407 所示。

步骤 1405、将第一列向量拆分为 X 个第二列向量，将第一行向量拆分为 X 个第二行向量。第一列向量和第一行向量中包括的元素值的精度为第一精度，第二列向量和第二行向量包含的元素值的精度为第二精度，第一精度高于第二精度，X 为大于或者等于 2 的整数。

35 本步骤请参阅上述矩阵计算装置的实施例中，图 10 和图 11 对应的示例中，格式转换单元 405 对整型数值进行拆分和格式转换单元 405 对浮点型数值进行拆分的具体说明，此处不赘述。

40 步骤 1406、计算 X 个第二列向量和 X 个第二行向量的向量外积，得到 X^2 个第四矩阵，第四矩阵包括第四元素值及第四元素值的位置坐标，第四元素值的位置坐标包括第一元素值的行坐标和第二元素值的列坐标，第四元素值的精度为第一精度。

本步骤请参阅上述矩阵计算装置的实施例中，图 12 对应的示例中，向量外积处理引擎 401 所执行的功能的说明，此处不赘述。

步骤 1407、根据第四元素值的位置坐标的索引，将 X^2 个第四矩阵中位置坐标相同的第四元素值累加，得到中间结果矩阵，中间结果矩阵中的第三元素值的精度为第一精度。

5 步骤 1408、根据第三元素值的位置坐标的索引，将 N 个中间结果矩阵中位置坐标相同的第三元素值进行累加，得到结果矩阵。

步骤 1407 和步骤 1408 请参阅上述矩阵计算装置的实施例中，图 5E，图 6 和图 8 对应的示例中累加器 402 执行的功能，此处不赘述。

本申请实施例中，在计算两个向量的向量外积时，可以进一步的将第一精度（高精度）的向量拆分成多个第二精度（低精度）的向量，然后对低精度的向量进行向量外积计算，从而使得第一精度的向量的外积可由多个第二精度的向量的外积结果做累积而得，且精度不会损失。

在本申请的一种实施例中，提供了一种矩阵计算的电路，所述矩阵计算电路用于上述方法实施例中步骤 1301-步骤 1303 中的一个或者多个步骤，或者，用于执行步骤 1401-步骤 15 1408 中的一个或者多个步骤。在实际应用中，该矩阵计算电路可以是 ASIC、FPGA 或者逻辑电路等。

在本申请的另一种实施例中，还提供一种矩阵计算系统或者芯片，该系统或者芯片的结构可以如图 3 所示，包括：处理器（以中央处理器为例）1 和矩阵计算装置 1。其中，处理器 1 用于向矩阵计算装置 1 发送计算指令，矩阵计算装置 2 用于执行上述方法实施例中的步骤 20 1301-步骤 1303 中的一个或者多个步骤，或者步骤 1401-步骤 1408 中的一个或者多个步骤。

在本申请的又一种实施例中，提供一种矩阵计算设备，该设备的结构可以如图 2 所示，该设备具体可以为 PCIe 卡、SoC、处理器，以及包括上述硬件的服务器等。参见图 2，该装置包括：存储器 201、处理器 202、通信接口 203 和总线 204。其中，通信接口 203 可以包括 25 输入接口和输出接口。

其中，处理器 202 可以被配置为执行上述方法实施例中的步骤 1301-步骤 1303 中的一个或者多个步骤，或者步骤 1401-步骤 1408 中的一个或者多个步骤。在一些可行的实施例中，处理器 202 可以包括矩阵计算单元，该矩阵计算单元可用于支持该处理器执行上述方法实施例中的一个或者多个步骤。在实际应用中，该矩阵计算单元可以是 ASIC、FPGA 或者逻辑电路等，当然该矩阵计算单元也可以通过软件实现，本申请实施例对此不作具体限制。 30

需要说的是，本申请实施例提供的上述矩阵计算电路、矩阵计算系统和矩阵计算设备等的各组成部分分别用于实现相对应的前述方法实施例的各步骤的功能，由于在前述的方法实施例中，已经对各步骤进行了详细说明，在此不再赘述。

上述实施例，可以全部或部分地通过软件、硬件、固件或其他任意组合来实现。当使用软件实现时，上述实施例可以全部或部分地以计算机程序产品的形式实现。所述计算机程序产品包括一个或多个计算机指令。在计算机上加载或执行所述计算机程序指令时，全部或部分地产生按照本申请实施例所述的流程或功能。所述计算机可以为通用计算机、专用计算机、计算机网络、或者其他可编程装置。所述计算机指令可以存储在计算机可读存储介质中，或者从一个计算机可读存储介质向另一个计算机可读存储介质传输，例如，所述计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线（例如同轴电缆、 40

光纤、数字用户线（DSL）或无线（例如红外、无线、微波等）方式向另一个网站站点、计算机、服务器或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存取的任何可用介质或者是包含一个或多个可用介质集合的服务器、数据中心等数据存储设备。所述可用介质可以是磁性介质（例如，软盘、硬盘、磁带）、光介质（例如，DVD）、或者半导体介质。半导体介质可以是固态硬盘（solid state drive, SSD）。

以上所述，以上实施例仅用以说明本申请的技术方案，而非对其限制；尽管参照前述实施例对本申请进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本申请各实施例技术方案的范围。

10

15

20

权 利 要 求 书

1. 一种矩阵计算装置，其特征在于，包括：向量外积处理引擎和累加器；

5 所述向量外积处理引擎，用于计算 N 个第一列向量和 N 个第一行向量的向量外积，得到 N 个中间结果矩阵，其中，所述第一列向量包括第一元素值及所述第一元素值的行坐标，所述第一行向量包括第二元素值及第二元素值的列坐标，所述中间结果矩阵包括第三元素值及所述第三元素值的位置坐标，所述位置坐标包括所述第一元素值的行坐标和所述第二元素值的列坐标；所述 N 个第一列向量是由压缩格式的第一矩阵转换得到的，所述 N 个第一行向量是由压缩格式的第二矩阵转换得到的，所述 N 为大于或者等于 1 的整数；

10 所述累加器，用于根据所述第三元素值的位置坐标的索引，将 N 个所述中间结果矩阵中位置坐标相同的第三元素值进行累加，得到结果矩阵。

2. 根据权利要求 1 所述的装置，其特征在于，所述 N 个所述中间结果矩阵至少包括第一中间结果矩阵和第二中间结果矩阵，所述第一中间结果矩阵中第三元素值的位置坐标为第一位置坐标，所述第二中间结果矩阵中第三元素值的位置坐标为第二位置坐标；

15 所述累加器还具体用于：

按照 N 个所述中间结果矩阵的产生顺序，将所述第一中间结果矩阵中第三元素值按照所述第一位置坐标写入缓存器中的对应位置；

20 根据所述第二中间结果矩阵中第三元素值的第二位置坐标，读取所述第二位置坐标在所述缓存器中对应位置的缓存值，将所述第二中间结果矩阵中第三元素值和所述缓存值进行累加，得到非压缩格式的结果矩阵。

3. 根据权利要求 2 所述的装置，其特征在于，所述装置还包括矩阵压缩单元；

所述矩阵压缩单元，用于压缩所述非压缩格式的结果矩阵，得到压缩格式的结果矩阵。

4. 根据权利要求 1 所述的装置，其特征在于，所述累加器还具体用于：

将所述 N 个中间结果矩阵中第三元素值按照所述第三元素值的位置坐标进行排序；

25 将排序后的所述 N 个中间结果矩阵中的位置坐标进行比对，将位置坐标相同的第三元素值相加，并删除零元素值的位置坐标，得到压缩格式的结果矩阵。

5. 根据权利要求 1 所述的装置，其特征在于，所述装置还包括格式转换单元；

所述格式转换单元，用于获取所述第一矩阵和所述第二矩阵，将所述第一矩阵转换成 N 个第一列向量，将所述第二矩阵转换成 N 个第一行向量。

30 6. 根据权利要求 5 所述的装置，其特征在于，所述装置还包括格式转换单元；

所述格式转换单元，还用于获取第五矩阵和第六矩阵，对所述第五矩阵进行格式转换，得到所述第一矩阵，对所述第六矩阵进行格式转换，得到所述第二矩阵，其中，所述第五矩阵和所述第六矩阵中至少有一个矩阵为非压缩格式的矩阵。

7. 根据权利要求 1-6 中任一项所述的装置，其特征在于，所述第一列向量和所述第一行向量中包括的元素值的精度为第一精度；

所述向量外积处理引擎，还用于计算 X 个第二列向量和 X 个第二行向量的向量外积，得到 X^2 个第四矩阵，所述第四矩阵包括第四元素值及所述第四元素值的位置坐标，所述第四元素值的位置坐标包括所述第一元素值的行坐标和所述第二元素值的列坐标，所述第四元素值的精度为第一精度；其中，所述 X 个第二列向量是由所述第一列向量拆分后得到的，所述 X

个第二行向量是由所述第一行向量拆分后得到的,所述第二列向量和所述第二行向量包含的元素值的精度为第二精度,所述第一精度高于所述第二精度, X 为大于或者等于2的整数;

所述累加器,还用于根据所述第四元素值的位置坐标的索引,将所述 X^2 个第四矩阵中位置坐标相同的第四元素值累加,得到所述中间结果矩阵,所述中间结果矩阵中的第三元素值的精度为所述第一精度。

8. 一种矩阵计算方法,其特征在于,所述方法应用于矩阵计算装置,所述方法包括:

获取第一计算指令,所述第一计算指令包括 N 个第一列向量和 N 个第一行向量;

计算 N 个第一列向量和 N 个第一行向量的向量外积,得到 N 个中间结果矩阵,其中,所述
10 所述第一列向量包括第一元素值及所述第一元素值的行坐标,所述第一行向量包括第二元素值及第二元素值的列坐标,所述中间结果矩阵包括第三元素值及所述第三元素值的位置坐标,所述位置坐标包括所述第一元素值的行坐标和所述第二元素值的列坐标;所述 N 个第一列向量是由压缩格式的第一矩阵转换得到的,所述 N 个第一行向量是由压缩格式的第二矩阵转换得到的,所述 N 为大于或者等于1的整数;

根据所述第三元素值的位置坐标的索引,将 N 个所述中间结果矩阵中位置坐标相同的第
15 三元素值进行累加,得到结果矩阵。

9. 根据权利要求8所述的方法,其特征在于,所述 N 个所述中间结果矩阵至少包括第一中间结果矩阵和第二中间结果矩阵,所述第一中间结果矩阵中第三元素值的位置坐标为第一位置坐标,所述第二中间结果矩阵中第三元素值的位置坐标为第二位置坐标;

所述根据所述第三元素值的位置坐标的索引,将 N 个所述中间结果矩阵中位置坐标相同的
20 的第三元素值进行累加,得到结果矩阵,包括:

按照 N 个所述中间结果矩阵的产生顺序,将所述第一中间结果矩阵中第三元素值按照所述
第一位置坐标写入缓存器中的对应位置;

根据所述第二中间结果矩阵中第三元素值的第二位置坐标,读取所述第二位置坐标在所
25 述缓存器中对应位置的缓存值,将所述第二中间结果矩阵中第三元素值和所述缓存值进行累加,得到非压缩格式的结果矩阵。

10. 根据权利要求9所述的方法,其特征在于,所述方法还包括:

压缩所述非压缩格式的结果矩阵,得到压缩格式的结果矩阵。

11. 根据权利要求8所述的方法,其特征在于,所述根据所述第三元素值的位置坐标的索引,将 N 个所述中间结果矩阵中位置坐标相同的第三元素值进行累加,得到结果矩阵,包
30 括:

将所述 N 个中间结果矩阵中第三元素值按照所述第三元素值的位置坐标进行排序;

将排序后的所述 N 个中间结果矩阵中的位置坐标进行比对,将位置坐标相同的第三元素
值相加,并删除零元素值的位置坐标,得到压缩格式的结果矩阵。

12. 根据权利要求8所述的方法,其特征在于,在所述获取第一计算指令之前,所述方
35 法还包括:

获取第二计算指令,所述第二计算指令包括所述第一矩阵和所述第二矩阵;

将所述第一矩阵转换成 N 个第一列向量,将所述第二矩阵转换成 N 个第一行向量。

13. 根据权利要求12所述的方法,其特征在于,在所述获取第二计算指令之前,所述
方法还包括:

40 获取第三计算指令,所述第三计算指令包括第五矩阵和第六矩阵,其中,所述第五矩阵

和所述第六矩阵中至少有一个矩阵为非压缩格式的矩阵；

对所述第五矩阵进行格式转换，得到压缩格式的所述第一矩阵，对所述第六矩阵进行格式转换，得到所述第二矩阵。

14. 根据权利要求 8-13 中任一项所述的方法，其特征在于，所述第一列向量和所述第一行向量中包括的元素值的精度为第一精度；

所述计算 N 个第一列向量和 N 个第一行向量的向量外积，得到 N 个中间结果矩阵，包括：

- 计算 X 个第二列向量和 X 个第二行向量的向量外积，得到 X^2 个第四矩阵，所述第四矩阵包括第四元素值及所述第四元素值的位置坐标，所述第四元素值的位置坐标包括所述第一元素值的行坐标和所述第二元素值的列坐标，所述第四元素值的精度为第一精度；其中，所述 X 个第二列向量是由所述第一列向量拆分后得到的，所述 X 个第二行向量是由所述第一行向量拆分后得到的，所述第二列向量和所述第二行向量包含的元素值的精度为第二精度，所述第一精度高于所述第二精度， X 为大于或者等于 2 的整数；

所述方法还包括：

- 根据所述第四元素值的位置坐标的索引，将所述 X^2 个第四矩阵中位置坐标相同的第四元素值累加，得到所述中间结果矩阵，所述中间结果矩阵中的第三元素值的精度为所述第一精度。

15. 一种矩阵计算的电路，其特征在于，所述矩阵计算电路用于执行如权利要求 8-14 中任一项所述的方法。

16. 一种矩阵计算的系统，其特征在于，所述系统包括处理器和矩阵计算的装置，所述处理器用于向所述矩阵计算的装置发送计算指令，所述矩阵计算的装置用于执行如权利要求 8-14 中任一项所述的方法。

17. 一种芯片，其特征在于，所述芯片包括处理器，所述处理器中集成有矩阵计算装置，所述矩阵计算装置用于执行如权利要求 8-14 中任一项所述的矩阵计算方法。

18. 一种矩阵计算的设备，其特征在于，所述设备包括如权利要求 16 所述的矩阵计算的系统，或者如权利要求 17 所述的芯片。