# 一种基于脉动阵列的二值复数神经网络卷积计算加速装置与方法

## 技术领域

[0001] 本发明涉及人工智能领域，具体涉及一种基于脉动阵列的二值复数神经网络卷积计算加速装置与方法。

## 背景技术

[0002] 近年来，随着人工智能的飞速发展，深度神经网络（Deep Neural Networks，DNN）在诸多领域都有着广泛应用，如目标检测、人脸识别、图像视频超分处理、自动驾驶以及语音识别等。DNN算法具有存储密集和计算密集两大特性，因为其中包含了大量的权重数据和特征图数据之间的卷积操作，使得目前的DNN模型无法在边缘侧充分地训练或者推理。

[0003] 在进行卷积操作时，一般会将一张特征图的卷积操作转换成多个小型的矩阵乘法；如果采用传统的计算系统，即从内存中读取特征图矩阵数据和卷积核权重矩阵数据，完成计算后写回存储器，再继续下一批数据的计算，这种方法会导致数据复用率低，内存访问次数多、耗时久，计算效率低等问题。所以在人工智能芯片领域多采用脉动阵列的计算方式，以权重固定型脉动阵列完成矩阵乘法为例，其中A为输入特征图矩阵，B为卷积核数据矩阵，C为输出特征图矩阵，计算方法为：首先将权重矩阵数据在脉动阵列中自上而下传播，传播完成后将权重数据固定于每个脉动阵列单元中；再将输入特征图矩阵数据按行依次间隔一个周期输入脉动阵列，并自左向右传播；输入数据与权重数据进行乘法计算，乘法结果与上一级计算单元的部分和输出进行累加计算，所得新的部分和结果向下传播给下一级计算单元；最后一级计算单元将最终的部分和结果传入累加器中。

[0004] 二值复数神经网络有效地结合二值神经网络和复数神经网络，其中每层的输入特征图数据、权重数据和输出特征图数据都使用二值化的复数值来表示，即{1+i,1-i,-1+i,-1-i}中某一个，在进行卷积计算时，原先的全精度乘法计算也相应地映射为单比特的同或运算（xnor）外加一个计“1”（popcount）操作，在数据存储时，使用双比特替代32比特的单精度浮点数或16比特的半精度浮点，该网络具有二值神经网络的计算效率、低硬件开销、高能效比、高鲁棒性、高精度等多种优势。

[0005] 假设二值复数为激活输入，二值复数为权重，是全精度复数输出。二值复数的点积运算遵循复数计算规则，形式如下式，其中，，， ∈ {+1，-1}。一个二值复数点积运算需要结合4个二值点积和2个额外的实数加法。

## 发明内容

[0006] 为解决深度神经网络高硬件开销、计算效率低、无法在资源受限的边缘侧设备中部署等问题，实现低功耗、低成本、高能效比的边缘智能设备的目的，本发明采用如下的技术方案：

一种基于脉动阵列的二值复数神经网络卷积计算加速装置，装置中包括两个移位寄存器阵列、针对二值复数设计的基本运算单元PE组成的脉动阵列、相邻PE之间的寄存器和一个带累加器的SRAM存储单元。其中基本运算PE包含了权重数据输入接口（in\_d）、权重数据输出接口（out\_c）、特征图数据输入接口（in\_a）、特征图数据输出接口（out\_a）、上级基本运算单元PE部分和输入接口（in\_b）、部分和输出接口（out\_b）、控制信号输入接口、控制信号输出接口、存储权重数据的寄存器C1和C2、进位保留加法器（Carry Saved Adder，CSA）、用于二值复数popcount(xnor)计算的BC\_POPC(XNOR)模块、三个二选一数字选择器。

[0007] 进一步地，阐述加速装置各组件之间的电路连接关系，脉动阵列中的第一行基本运算单元的权重数据输入接口（in\_d）用于接收来自片上存储的卷积核权重矩阵；第一列基本运算单元特征图数据输入接口（in\_a）与左侧移位寄存器阵列连接，用于接收输入特征图矩阵；最后一行基本运算单元的部分和输出接口（out\_b）与底侧移位寄存器阵列连接，移位寄存器阵列将各列基本运算单元的输出结果进行打拍同步后送往带累加器的SRAM存储单元；每个基本运算单元的权重数据输出接口（out\_d）通过寄存器打一拍与同一列下一行的基本运算单元的权重数据输入接口（in\_d）连接，部分和接口（out\_b）与同一列下一行的基本运算单元的部分和输入接口（in\_b）直接连接，特征图数据输出接口（out\_a）和控制信号输出端口通过寄存器打一拍后与同一行下一列的基本运算单元的特征图数据输入接口（in\_a）和控制信号输入端口连接，故每个基本运算单元仅与相邻基本运算单元进行数据交互。

[0008] 进一步地，阐述加速装置内部的数据传输机制，加速装置内部通过握手机制来控制数据的传输，一组握手信号包括valid信号和ready信号，在数据的交互过程中，当valid和ready信号同时高有效时，数据在时钟上升沿传输。

[0009] 进一步地，阐述移位寄存器阵列内部的连接关系，左侧移位寄存器阵列模块的输入可分为握手信号接口、控制信号接口、输入数据信号接口；将输入信号分为16组分别送往脉动阵列的各行的第一列基本运算单元，每32比特的数据信号对应一组握手信号和1比特控制信号；其中第一组输入信号通过一个寄存器打一拍后与第一行第一列的基本运算单元特征图数据输入接口连接，第二组输入信号通过两个寄存器打一拍后与第二行第一列的基本运算单元特征图数据输入接口连接，依次类推，第16组输入信号通过16个寄存器打16拍后与第16行第一列的基本运算单元特征图数据输入接口连接，该移位寄存器阵列可以保证每组输入数据依次延迟一个周期后进入脉动阵列。底侧移位寄存器阵列模块结构与左侧移位寄存器阵列模块结构类似，输入可分为16组握手信号接口和输出数据信号，输入出为一个握手信号组和输出数据信号组，第一列最后一行的基本运算单元的部分和接口（out\_b）经过16个移位寄存器打16拍后与第一组握手信号和输出数据信号接口连接，第二列最后一行的基本运算单元的部分和接口（out\_b）经过15个移位寄存器打15拍后与第二组握手信号和输出数据信号接口连接，以此类推，该移位寄存器阵列可以保证每组输入数据打拍后同步进入带累加器的SRAM存储器中。

[0010] 进一步地，阐述基本运算单元内部的连接关系，在基本运算单元内部，特征图数据输入接口（in\_a）与特征图数据输出接口（out\_a）直接连接；权重数据输入接口（in\_d）经过双缓存结构通过数字选择器Mux3与权重数据输出接口（out\_c）连接，双缓存结构由数字选择器Mux0、Mux1和寄存器C1、C2组成；数字选择器Mux2的输入端分别与寄存器C1、C2连接，选择信号端与控制信号PROP连接，输出端与BC\_POPC(XNOR)模块连接；BC\_POPC(XNOR)模块的另一个输入数据端与特征图数据输入接口（in\_a）连接，权重数据和特征图输入数据进行popcount(xnor)计算后输出；进位保留加法器的两个输入端分别与BC\_POPC(XNOR)模块的输出端和上级基本运算单元PE部分和输入接口（in\_b）连接，输出端与部分和输出接口（out\_b），将上级基本运算单元部分和计算结果与BC\_POPC(XNOR)模块的输出结果累加后输出；

所述双缓冲结构主要用于缓冲输入数据便于后续的计算，且通过双缓冲可以在当前计算周期未完成的情况下将先前的计算结果从脉动阵列输出，实现了non-stop计算，极大的提高了计算效率。针对两条数据通路，我们在电路的输入端和输出端各添加两个数字选择器Mux0、Mux1和Mux3，将控制信号PROP作为三个数字选择器的选择信号，当PROP为高电平时，该结构利用寄存器C1所在通路进行存储和传播权重数据，当PROP为低电平时，该结构利用寄存器C2所在通路进行存储和传播权重数据。

所述基本运算单元PE内部通过数字选择器Mux2以及控制信号PROP来选择寄存器C1、C2中哪一个寄存器的权重数据用于计算，当PROP为高电平时，选择寄存器C2中存储的权重数据送往BC\_POPC(XNOR)模块进行二值复数乘法计算，当PROP为低电平时，选择寄存器C1中存储的权重数据送往BC\_POPC(XNOR)模块进行二值复数乘法计算。

[0011] 进一步地，所述BC\_POPC(XNOR)模块中含有四个16比特的popc(xnor)计算单元、两个加法器，特征图数据输入接口、权重数据输入接口以及特征图数据输出接口。来自输入接口的特征图数据和权重数据均为32比特，进入BC\_POPC(XNOR)模块后首先分别分解为高16位与低16位两个部分，其中高16位代表多通道的二值复数的实部数据的集和，低16位代表多通道的二值复数的虚部数据的集和；计算过程遵循复数的乘法法则，特征图数据的实部分别送往第一和第三个popc(xnor)计算单元，特征图数据的虚部分别送往第二和第四个popc(xnor)计算单元，同理分配权重数据的实部和虚部。完成popcount(xnor)计算后，将第一个与第二个popc(xnor)计算单元的计算结果求差，得到特征图输出数据的实部，将第三个与第四个popc(xnor)计算单元的计算结果求和，得到特征图输出数据的虚部。其中popc(xnor)计算单元用于实现popcount(xnor)计算，即将两个16比特数据按位同或后得到中间计算结果，再将中间结果通过并行结构按位相加得到中间结果中“1”的个数，作为最终结果输出。

[0012] 进一步地，阐述含累加器的SRAM存储器结构，该SRAM存储中含有写端口、读端口、双端口RAM、一至二数据分发器Demux、累加器adder、三个流水线队列。其中写端口中分为写地址输入接口、写数据输入接口、握手信号输入接口，读端口分为读地址输入接口、读数据输出接口、握手信号输入接口。写端口信号通过流水线队列pipeline0处理后与数据分发器Demux的输入端连接；Demux的两个输出端分别与双端口RAM写数据端口和累加器adder的操作数端口（op2）连接；累加器adder的操作数端口（op1）与双端口RAM的读数据接口连接，累加器的输出端口（sum）与双端口RAM的写数据端口连接。读数据端口通过流水线队列pipeline1与双端口RAM的读端口连接，双端口RAM中该读地址对应的读数据通过流水线队列pipeline2输出。

所述流水线队列pipeline0中包含译码电路，会对写地址进行译码，通过地址信号的第30比特，判断该组信号中写数据是直接覆盖该地址的原始数据还是与该地址的原始数据累加；

所述累加器adder会将需要累加的写数据与双端口RAM中写地址对应的原始数据进行累加，并将结果从数据输出端口（sum）输出、写回双端口RAM中写地址对应存储单元。

[0012] 基于上述加速装置，提出一种基于脉动阵列的二值复数神经网络卷积计算方法，包括如下步骤：

S1，获取多个卷积核权重数据和输入特征图数据；

S2，权重数据输入接口（in\_d）中握手信号握手成功后，在控制信号PROP的控制下，将权重数据通过权重输入通道分发到脉动阵列的各列基本运算单元PE，通过双缓存结构广播并保留在寄存器中；

S3，特征图数据输入接口（in\_a）中握手信号握手成功后，特征图数据通过左侧移位寄存器阵列分组打拍，沿着行方向在脉动阵列中向后传播，同时基本运算单元对保存的权重数据和输入的特征图数据进行popcount(xnor)计算和累加操作，包括如下过程：

S31，左侧移位寄存器阵列对特征图数据进行分组并打拍，输入数据按行依次延迟一个周期进入脉动阵列；

S32，第一列基本运算单元PE在接收到特征图输入数据时，在控制信号PROP的控制下，选择双缓存接口中某一个寄存器C1或者C2保存的权重数据与特征图数据进行popcount(xnor)计算，同时特征图输入数据通过特征图输出接口（out\_a）向同一行的下一列基本运算单元PE传播；

S33，BC\_POPC(XNOR)模块将计算结果送往进位保留加法器CSA，与上级基本运算单元PE的部分和输入接口（in\_b）的输入进行累加，并将新的部分和传输给同一列的下级基本运算单元PE；

S34，最后一行的基本运算单元PE将最终的部分和结果，通过底侧移位寄存器阵列打拍同步后传输给带累加器的SRAM存储器进行保存；

S4，带累加器的SRAM存储器根据写端口的地址信号判断是否进行累加，完成卷积计算后，通过读该SRAM存储器即可得到卷积的最终结果。

[0012] 进一步地，所述S1中的每个权重数据和每个特征图数据的数据量的组数分别对应于脉动阵列的每行基本运算单元PE的个数和每列基本运算单元PE的个数。每组数据为32比特，分别表示16个通道的二值复数的集合，其中高16位表示各通道的实部数据的集合，低16位表示各通道的虚部数据的集合。

[0013] 进一步地，所述S2中权重数据输入接口（in\_d）每次最多可输入16个卷积核的权重数据矩阵，如果卷积核个数大于16，则需要分批次进行卷积计算，并在带累加器的SRAM存储器中将部分和结果累加。

[0014] 进一步地，所述S3中二值复数的popcount(xnor)计算作用等同于整数或全精度浮点数的乘法计算，不过BC\_POPC(XNOR)模块仅使用了一系列的同或门和加法器，不仅节约了硬件开销，还提高了计算效率。

[0015] 本发明的优势和有益效果在于：

本发明的一种基于脉动阵列的二值复数神经网络卷积计算加速装置与方法，采用两比特的二值复数代替32比特的单精度浮点数或16比特的半精度浮点数，采用硬件开销小、计算简单的popcount(xnor)计算代替硬件开销大、计算复杂的浮点数乘法计算，并将多通道的二值复数合并为单通道进行计算，不仅节省了存储资源、减少了内存访问次数，还提高了计算效率；另外通过脉动阵列可以多次复用权重数据和输入特征图数据，一次内存访问，就可以完成多个卷积核的卷积计算，提高了加速装置的吞吐率和计算效率。综上所述在人工智能芯片领域采用该加速装置和计算方法，可以实现低功耗、低成本、高能效的边缘智能设备，解决了在资源受限的边缘侧设备中部署深度神经网络的难题。

## 具体实施方法

所述写端口所有接口信号到达存储器模块后，先通过流水线队列pipeline0，队列中对写地址信号进行译码，如果直接覆盖，则将握手、地址、数据信号直接送往双端口RAM；如果需要累加，则先读取双 端口RAM中该地址对应的数据，并

在权重固定模式下的脉动阵列，每完成一次矩阵乘加计算的结果将进入Accumulator并与目标地址保存的原值进行累加，这就需要在每个周期同时完成一次读数据任务、累加计算任务、写数据任务，虽然这三个任务可以通过寄存器缓存并使用状态机依次完成，但是在边缘智能设备这类实时性要求较高的应用场景中，这种方式显然是达不到算力要求的，所以我们采用了流水线形式将 三个任务并行执行，极大的缩短了算法的运行时间，因此accumulator也必须要使用双端SRAM实现。

## 附图说明

## 具体实施方式

## 参考文献

1. Bansal A, Borgnia E, Chu H M, et al. Cold diffusion: Inverting arbitrary image transforms without noise[J]. arXiv preprint arXiv:2208.09392, 2022.
2. Vaswani A , Shazeer N , Parmar N , et al. Attention Is All You Need[C]// arXiv. arXiv, 2017.