(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 106228238 A
(43)申请公布日 2016.12.14

(21)申请号 201610596159.3

(22)申请日 2016.07.27

(71)申请人 中国科学技术大学苏州研究院
地址 215123 江苏省苏州市工业园区独墅湖高教区仁爱路188号

(72)发明人 周学海 王超 余奇 周徐达 赵洋洋 李曦 陈香兰

(74)专利代理机构 苏州创元专利商标事务所有限公司 32103
代理人 范晴 丁浩秋

(51)Int.Cl.
G06N 3/06(2006.01)
G06N 3/08(2006.01)

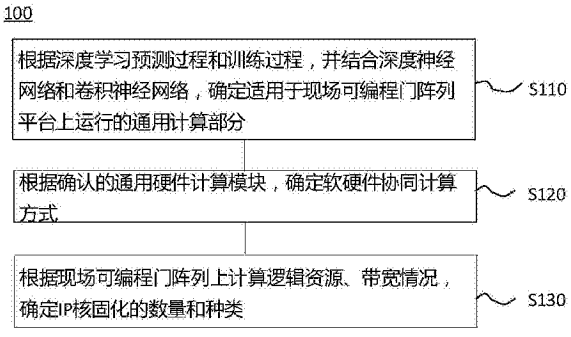
权利要求书2页 说明书10页 附图10页

(54)发明名称

现场可编程门阵列平台上加速深度学习算法的方法和系统

(57)摘要

本发明公开了一种现场可编程门阵列平台上加速深度学习算法的方法，现场可编程门阵列平台包括通用处理器、现场可编程门阵列以及存储模块，包括以下步骤：根据深度学习预测过程和训练过程，并结合深度神经网络和卷积神经网络，确定适用于现场可编程门阵列平台上运行的通用计算部分；根据确认的通用计算部分，确定软硬件协同计算方式；根据FPGA的计算逻辑资源、带宽情况，确定IP核固化的数量和种类，利用硬件运算单元，在现场可编程门阵列平台上进行加速。能够根据硬件资源快速设计出针对深度学习算法加速的硬件处理单元，处理单元相对于通用处理器有高性能、低功耗特点。



CN 106228238 A

1.一种现场可编程门阵列平台上加速深度学习算法的方法，其特征在于，现场可编程门阵列平台包括通用处理器、现场可编程门阵列以及存储模块，包括以下步骤：

S01：根据深度学习预测过程和训练过程，并结合深度神经网络和卷积神经网络，确定适用于现场可编程门阵列平台上运行的通用计算部分；

S02：根据确认的通用计算部分，确定软硬件协同计算方式；

S03：根据FPGA的计算逻辑资源、带宽情况，确定IP核固化的数量和种类，利用硬件运算单元，在现场可编程门阵列平台上进行加速。

2.根据权利要求1所述的现场可编程门阵列平台上加速深度学习算法的方法，其特征在于，所述通用计算部分包括前向计算模块，用于矩阵乘法计算和激励函数计算；权值更新模块，用于向量计算。

3.根据权利要求1所述的现场可编程门阵列平台上加速深度学习算法的方法，其特征在于，所述步骤S02包括以下步骤：

在软件端进行数据准备工作；

将卷积神经网络中卷积层卷积计算转化为矩阵乘法；

采用直接内存读取作为软硬件协同计算的数据通路。

4.根据权利要求1所述的现场可编程门阵列平台上加速深度学习算法的方法，其特征在于，所述步骤S03中确定IP核固化的数量和种类，包括：根据待执行的硬件任务，确定FPGA上固化的运算单元的种类；根据FPGA硬件逻辑资源和带宽情况，确定待执行硬件任务的处理单元的数量。

5.根据权利要求2所述的现场可编程门阵列平台上加速深度学习算法的方法，其特征在于，所述前向计算模块采用分片设计，将节点矩阵每一行内部按分片大小进行分片，权值参数矩阵每一列按照分片大小进行分片，按行将节点矩阵的每分片大小个数据与权值参数矩阵每一列对应的分片大小个数值进行点积运算，每一行计算完毕后将临时值累加得到最终结果。

6.根据权利要求5所述的现场可编程门阵列平台上加速深度学习算法的方法，其特征在于，所述分片大小为2的n次方，与运算单元的并行粒度保持一致。

7.一种用于加速深度学习算法的FPGA结构，其特征在于，包括：

分片处理结构，将前向计算模块的节点数据矩阵和权值参数矩阵进行分片，分时复用硬件逻辑；

激励函数线性近似实现结构，用于生成任意激励函数；

参数配置模块，用于配置处理单元的参数；

前向计算模块，包括单DMA缓存权值的前向计算硬件结构和双DMA并行读取的前向计算硬件结构；用于深度神经网络的前向计算、卷积神经网络卷积层和分类层的前向计算以及矩阵乘法操作，并且进行流水线优化至最大吞吐率；

权值更新模块，用于向量计算。

8.根据权利要求7所述的用于加速深度学习算法的FPGA结构，其特征在于，所述参数配置模块通过DMA传输配置参数数据对处理单元进行配置，包括：前向计算模块的工作模式配置和数据规模配置，数据规模配置包括节点数据规模配置、输入神经元规模配置和输出神经元规模配置；权值更新模块数据规模配置、工作模式配置和计算参数配置。

9.根据权利要求7所述的用于加速深度学习算法的FPGA结构，其特征在于，所述单DMA缓存权值的前向计算硬件结构包括：

单个DMA，负责数据读取、写回；

双寄存器缓冲区，交替读取数据或进行并行计算；BRAM组，缓存并保证数据并行读取；

同分片大小相等的浮点乘法器；

同分片大小相等输入的二叉加法树；

循环累加器，累加临时值保存至片上BRAM上；

激励函数计算模块，采用分段线性近似实现激励函数，计算系数缓存在片上BRAM；

所述双DMA并行读取的前向计算硬件结构包括：

神经元数据读取模块，配有DMA和FIFO缓存区，负责读取输入神经元节点数据；

权值参数数据读取模块，配有DMA和FIFO缓存区，负责读取权值参数数据；

同分片大小相等的浮点乘法器；

同分片大小相等输入的二叉加法树；

循环累加器，累加临时值保存至片上BRAM上；

激励函数计算模块，采用分段线性近似实现激励函数，计算系数缓存在片上BRAM。

10.根据权利要求7所述的用于加速深度学习算法的FPGA结构，其特征在于，所述权值更新模块，用于权值更新计算和输出层误差值的计算，并且进行流水线优化至最大吞吐率，包括：向量A数据读取模块和向量B数据读取模块，分别配有DMA和FIFO缓冲区，分别读取用于计算的两组向量值；计算模块，通过配置信息进行对应的向量计算；结果写回模块，配有DMA和FIFO缓冲区，将计算结果写回至宿主端内存。

现场可编程门阵列平台上加速深度学习算法的方法和系统

技术领域

[0001] 本发明涉及计算机硬件加速领域，具体地涉及一种现场可编程门阵列平台上加速深度学习算法的方法和系统。

背景技术

[0002] 深度学习在解决高级抽象认知问题上有着显著的成果，使机器学习上了一个新台阶。其不仅具有很高的科研价值，而且具有很强的实用性，致使无论学术界和工业界都十分青睐。然而，为了解决更加抽象、更加复杂的学习问题，深度学习的网络规模在不断增加，计算和数据的复杂也随之剧增，比如Google Cat系统网络具有10亿左右个神经元。高性能低能耗地加速深度学习相关算法成为科研和商业机构的研究热点。

[0003] 通常计算任务从表现方式上分两种：在通用处理器上，任务通常以软件代码的形式呈现，称为软件任务；在专用硬件电路上，充分发挥硬件固有的快速特性来代替软件任务，称为硬件任务。常见的硬件加速技术有专用集成电路ASIC(Application Specific Integrated Circuit)、现场可编程逻辑门阵列FPGA(Field Programmable Gate Array)和图形处理器GPU(Graphics Processing Unit)。ASIC是为特定用途设计开发的集成电路芯片，其具有高性能、低功耗、面积小等特点。通常相对于FPGA，ASIC运行更快、功耗更低，而且量化生产时也更便宜。虽然对于同一给定功能，FPGA所使用的晶体管要比ASIC要多，但FPGA简化了逻辑任务设计，设计周期要比ASIC短很多。此外，生产ASIC的掩膜成本很高，随着线宽的减小，掩膜成本成指数增长。FPGA作为适用不同功能的可编程标准器件，没有如此高额的研发成本，并且具有一定的灵活性。GPU适用于大量数据的并行计算，具有高带宽、高主频、高并行性特点，而且CUDA(Compute Unified Device Architecture)通用并行计算框架的提出，使开发者更方便、快捷地设计出高性能解决方案。但GPU的功耗较高，单个GPU的功耗往往要高于同期主流的CPU功耗，通常相对于FPGA要多几十倍甚至上百倍的能量消耗。

发明内容

[0004] 有鉴于此，本发明目的是：提供了一种现场可编程门阵列平台上加速深度学习算法的方法和系统，能够根据硬件资源快速设计出针对深度学习算法加速的硬件处理单元，处理单元相对于通用处理器有高性能、低功耗特点。

[0005] 本发明的技术方案是：

[0006] 一种现场可编程门阵列平台上加速深度学习算法的方法，其特征在于，现场可编程门阵列平台包括通用处理器、现场可编程门阵列以及存储模块，包括以下步骤：

[0007] S01：根据深度学习预测过程和训练过程，并结合深度神经网络和卷积神经网络，确定适用于现场可编程门阵列平台上运行的通用计算部分；

[0008] S02：根据确认的通用计算部分，确定软硬件协同计算方式；

[0009] S03：根据FPGA的计算逻辑资源、带宽情况，确定IP核固化的数量和种类，利用硬件运算单元，在现场可编程门阵列平台上进行加速。

[0010] 优选技术方案中，所述通用计算部分包括前向计算模块，用于矩阵乘法计算和激励函数计算；权值更新模块，用于向量计算。

[0011] 优选技术方案中，所述步骤S02包括以下步骤：

[0012] 在软件端进行数据准备工作；

[0013] 将卷积神经网络中卷积层卷积计算转化为矩阵乘法；

[0014] 采用直接内存读取作为软硬件协同计算的数据通路。

[0015] 优选技术方案中，所述步骤S03中确定IP核固化的数量和种类，包括：根据待执行的硬件任务，确定FPGA上固化的运算单元的种类；根据FPGA硬件逻辑资源和带宽情况，确定待执行硬件任务的处理单元的数量。

[0016] 优选技术方案中，所述前向计算模块采用分片设计，将节点矩阵每一行内部按分片大小进行分片，权值参数矩阵每一列按照分片大小进行分片，按行将节点矩阵的每分片大小个数据与权值参数矩阵每一列对应的分片大小个数值进行点积运算，每一行计算完毕后将临时值累加得到最终结果。

[0017] 优选技术方案中，所述分片大小为2的n次方，与运算单元的并行粒度保持一致。

[0018] 本发明又公开了一种用于加速深度学习算法的FPGA结构，其特征在于，包括：

[0019] 分片处理结构，将前向计算模块的节点数据矩阵和权值参数矩阵进行分片，分时复用硬件逻辑；

[0020] 激励函数线性近似实现结构，用于生成任意激励函数；

[0021] 参数配置模块，用于配置处理单元的参数；

[0022] 前向计算模块，包括单DMA缓存权值的前向计算硬件结构和双DMA并行读取的前向计算硬件结构；用于深度神经网络的前向计算、卷积神经网络卷积层和分类层的前向计算以及矩阵乘法操作，并且进行流水线优化至最大吞吐率；

[0023] 权值更新模块，用于向量计算。

[0024] 优选技术方案中，所述参数配置模块通过DMA传输配置参数数据对处理单元进行配置，包括：前向计算模块的工作模式配置和数据规模配置，数据规模配置包括节点数据规模配置、输入神经元规模配置和输出神经元规模配置；权值更新模块数据规模配置、工作模式配置和计算参数配置。

[0025] 优选技术方案中，所述单DMA缓存权值的前向计算硬件结构包括：

[0026] 单个DMA，负责数据读取、写回；

[0027] 双寄存器缓冲区，交替读取数据或进行并行计算；BRAM组，缓存并保证数据并行读取；

[0028] 同分片大小相等的浮点乘法器；

[0029] 同分片大小相等输入的二叉加法树；

[0030] 循环累加器，累加临时值保存至片上BRAM上；

[0031] 激励函数计算模块，采用分段线性近似实现激励函数，计算系数缓存在片上BRAM；

[0032] 所述双DMA并行读取的前向计算硬件结构包括：

[0033] 神经元数据读取模块，配有DMA和FIFO缓存区，负责读取输入神经元节点数据；

[0034] 权值参数数据读取模块，配有DMA和FIFO缓存区，负责读取权值参数数据；

[0035] 同分片大小相等的浮点乘法器；

[0036] 同分片大小相等输入的二叉加法树；

[0037] 循环累加器，累加临时值保存至片上BRAM上；

[0038] 激励函数计算模块，采用分段线性近似实现激励函数，计算系数缓存在片上BRAM。

[0039] 优选技术方案中，所述权值更新模块，用于权值更新计算和输出层误差值的计算，并且进行流水线优化至最大吞吐率，包括：向量A数据读取模块和向量B数据读取模块，分别配有DMA和FIFO缓冲区，分别读取用于计算的两组向量值；计算模块，通过配置信息进行对应的向量计算；结果写回模块，配有DMA和FIFO缓冲区，将计算结果写回至宿主端内存。

[0040] 与现有技术相比，本发明的优点是：

[0041] 本发明可以有效的加速深度学习算法，包括学习预测过程和训练过程，能够根据硬件资源快速设计出针对深度学习算法加速的硬件处理单元，处理单元相对于通用处理器有高性能、低功耗特点。

附图说明

[0042] 下面结合附图及实施例对本发明作进一步描述：

[0043] 图1是本发明实施例的现场可编程逻辑门阵列平台上加速深度学习方法的流程图；

[0044] 图2是卷积神经网络中卷积层的计算示意图；

[0045] 图3是本发明实施例的现场可编程逻辑门阵列平台上的前向计算硬件处理单元转换卷积层计算的示意图；

[0046] 图4是本发明实施例的现场可编程逻辑门阵列平台上的权值更新处理单元将数据矩阵转换成向量的示意图；

[0047] 图5是本发明实施例的现场可编程逻辑门阵列平台上软硬件协同计算的结构示意图；

[0048] 图6是本发明实施例的硬件处理单元资源使用和现场可编程逻辑门阵列平台资源以及应用情况固化数量和种类的示意图；

[0049] 图7是本发明实施例的前向计算处理单元数据分片处理的示意图；

[0050] 图8是本发明实施例的分段线性近似实现激励函数的示意图；

[0051] 图9是本发明实施例的异构多核可重构计算平台上单DMA预存权值矩阵的前向计算硬件处理单元的结构示意图；

[0052] 图10是本发明实施例的异构多核可重构计算平台上前向计算硬件处理单元中累加处理的结构示意图；

[0053] 图11是本发明实施例的异构多核可重构计算平台上前向计算硬件处理单元中分段近似sigmoid函数的结构示意图；

[0054] 图12是本发明实施例的异构多核可重构计算平台上单DMA预存权值矩阵的前向计算硬件处理单元的数据处理流程图；

[0055] 图13是本发明实施例的异构多核可重构计算平台上双DMA并行读取数据的前向计算硬件处理单元的结构示意图；

[0056] 图14是本发明实施例的异构多核可重构计算平台上双DMA并行读取数据的前向计算硬件处理单元的数据处理流程图；

[0057] 图15是本发明实施例的异构多核可重构计算平台上权值更新硬件处理单元的结构示意图；

[0058] 图16是本发明实施例的异构多核可重构计算平台上权值更新硬件处理单元的数据处理流程图；

[0059] 图17是本发明实施例的异构多核可重构计算平台上深度学习加速器的可能一个应用场景及框架示意图。

具体实施方式

[0060] 以下结合具体实施例对上述方案做进一步说明。应理解，这些实施例是用于说明本发明而不限于限制本发明的范围。实施例中采用的实施条件可以根据具体厂家的条件做进一步调整，未注明的实施条件通常为常规实验中的条件。

[0061] 实施例：

[0062] 本发明实施例中的现场可编程门阵列平台指同时集成通用处理器(General Purpose Processor，简称为“GPP”)，和现场可编程门阵列(Field Programmable Gate Arrays，简称为“FPGA”)芯片的计算系统，其中，FPGA和GPP之间的数据通路可以采用PCI-E总线协议、AXI总线协议等。本发明实施例附图数据通路采用AXI总线协议为例说明，但本发明并不限于此。

[0063] 图1为本发明实施例的现场可编程门阵列平台加速深度学习算法的方法100的流程图。该方法100包括：

[0064] S110，根据深度学习预测过程和训练过程，其中训练过程包含本地预训练过程和全局训练过程，并结合深度神经网络和卷积神经网络，确定适用于现场可编程门阵列平台上运行的通用计算部分；

[0065] S120，根据确认的通用硬件计算模块，确定软硬件协同计算方式；

[0066] S130，根据现场可编程门阵列上计算逻辑资源、带宽情况，确定IP核固化的数量和种类。

[0067] 下文中将结合图2至图4，对本发明实施例加速深度学习通用计算部分的方法进行详细描述。

[0068] 图2为卷积层计算的示意图，假设输入特征图个数为4，卷积核大小为3x3，则将4个卷积结算结果累加求和后，经过激励函数处理即可得到输出特征图的值。从计算整体结构上看，卷积层的基本计算方式和深度神经网络隐层计算类似，只要通过调整卷积核参数序列便可将这里使用的卷积计算变化成点积计算。具体调整方式为：1)、将输入特征图从上至下、按行依次填充至一行，如图3左边行所示；2)将卷积矩阵核逆时针旋转180度后，从上至下、按行循序写入权值矩阵的一列，图3中间那一列所示，将原有卷积核a至卷积核d依次逆时针旋转180度后，变成a9～a1、b9～b1、……d9～d1，在循序填充至一列中。所以，针对卷积层预测过程，其基本计算可转换成与深度神经网络隐层相同的方式，即矩阵乘法计算加上激励函数处理，不过需要多付出数据转换的代价。

[0069] 在深度学习训练过程中，除了需要大量的矩阵乘法计算还需要大量的向量计算，在进行向量计算时需要将矩阵数据转换成向量数据，如图4所示，将数据每一行循序组成一个向量进行向量计算。

[0070] 因此，结合图2至图4，本发明实例将深度学习预测过程和训练过程的通用计算部分归结为矩阵乘法计算、激励函数计算和大量的向量计算。

[0071] 图5为本发明实例采用的软硬件协同计算的结构框架图200。该结构包括：

[0072] Processing System(简称PS)210，作为整个系统的控制端，包含CPU和Memory。CPU作为宿主端，运行软件端代码，并将加速任务offload至PL端进行工作。此外，CPU作为可控制PL端各IP核(intellectual property core，这里代表各硬件运算单元)的工作状态和数据读取等等；

[0073] 可编程逻辑Programming Logic(简称PL)220，为整个系统的硬件加速部件FPGA芯片。可以根据不同加速任务在FPGA芯片上固化IP核来实现对算法的加速。系统由PS端根据具体算法调度选择不同的IP Core进行并行计算，也可以将宿主端软件任务和FPGA端硬件任务进行并行计算；

[0074] 数据总线(Data Bus)230，负责整个系统PS端和PL端数据传输；

[0075] 控制信号总线(Control Bus)240，负责整个系统PS端和PL端控制信号的传输。

[0076] 图6为基于FPGA设计的加速器总体结构2000，结构包括：

[0077] 系统控制器2100，负责控制各硬件运算单元的执行状态、数据传输以及程序调度。并且负责运行深度学习非通用的计算部分，数据初始化和硬件运算单元(或称为IP核)的初始化任务；

[0078] 内存2200，负责存储深度学习网络参数以及原始输入数据，这里要求数据存储的物理地址为连续的，方便DMA进行数据传输；

[0079] 数据总线协议2300，AXI-Stream协议允许无限制的数据突发传输，为高性能数据传输协议；

[0080] 控制总线协议2400，AXI-Lite是一种轻量级的地址映射单次传输协议，适用于硬件运算单元的控制信号传输；

[0081] 数据互联2500，数据通路互联；

[0082] 控制互联2600，控制信号线路互联；

[0083] 直接内存存取DMA2700，负责加速器和内存间的数据传输，每个硬件处理单元均配备一个DMA来并行读取数据；

[0084] PE(Proccesing Element)2800作为每个加速器的计算单元，内部可固化1个前向计算运算单元或者1个权值更新运算单元或者两者均包含。由于FPGA具有可编程性和可重构性，这里PE的数量可根据具体FPGA芯片的资源带宽情况动态配置，这样在不改变运算单元硬件设计下可以充分利用硬件的计算资源，保证硬件发挥最高性能。

[0085] 上文中结合图1至图6，详细描述了本发明实施例加速深度学习算法的方法，下面将介绍本发明实施例的硬件结构。

[0086] 图7为采用分片计算思想设计前向计算运算单元，假设分片的大小为16，将节点矩阵每一行内部按16进行分片，权值参数矩阵按照每一列16个元素进行分片。按行将节点矩阵的每16个数据与权值参数矩阵每一列对应的16个数值进行点积运算，待每一行计算完毕后再将这些临时值累加即可得到最终结果。此种方法不仅充分利用了数据局部性，而且减少了固化并行执行单元所需的资源情况，并降低了硬件所需数据带宽，让单个运算单元可以实现任意规模的矩阵乘法计算。

[0087] 为了保持高吞吐率，分片的大小应与运算单元内部设计相配合，同并行粒度保持一致，在矩阵乘法运算时，可以将分片设定为2的n次方，来充分发挥二叉树的累加性能。由于分片大小与并行粒度有关，理论上来说分片越大，并行度越高，运算单元的性能也会越好，所以在硬件资源和带宽允许的情况下，选择最大的2^n作为运算单元的分片大小。

[0088] 图8是本发明实例中对激励函数进行硬件实现的示意图。本发明实例采用分段线性近似来实现S型激励函数，将函数按X轴划分为若干等值间隔，每个间隔内按$Y=a_i*X+b_i, X\in[x_i, x_{i+1})$所示进行线性近似，其中$x_{i+1}-x_i$为近似的间隔大小。每当需要计算激励函数时，首先按照X值寻找其所在的区间并计算其对应的a_i和b_i相对于基地址的偏移量，进行乘加运算后，即可近似得到Y值。这种实现方式有两点好处：1)、可实现任意的S型激励函数或线性函数，而且无需更改任何硬件设计，仅需要更换系数a和系数b所存储的数值即可；2)、误差极小，当近似区间降低时，误差可以达到可以忽略，而代价仅仅是增加用于存储系数a和系数b的BRAM。而且深度学习计算本身对数据的精确度的要求并不是很高或者说一定程度的精度损失并不影响数据结果。

[0089] 图9是本发明实施例的现场可编程门阵列平台上单DMA预存权值矩阵的硬件结构的示意性框图3000，该硬件结构针对FPGA内部BRAM资源比较充足时，预先缓存权值矩阵数据在片上BRAM进行前向计算。结构包括：

[0090] 数据读取模块3100，配有DMA和FIFO缓存区，数据位宽为32位，负责读取权值参数缓存在片上BRAM上以及读取神经元节点数据。

[0091] 片上BRAM3200，缓存权值参数数据。以分片大小为16为例，将权值矩阵按行以16为循环存入不同的BRAM上，即i％16加上BRAM的基地址作为寻址方式，从而保证在进行16个并行乘法时从不同的BRAM并行读取数据。

[0092] 双寄存器缓存3300，这里每个寄存器包含16个寄存器用于存储输入神经元数据，通过替进行缓存数据和进行并行计算。不过这里需要注意的是：将缓存区填满所需的时间要低于这些数据计算所需的时间，这样才能保证缓冲区数据读取的时间被计算所需时间所覆盖，并确保结果的正确性。

[0093] 并行浮点乘法3400，将权值参数数据和神经元数据进行并行乘法计算，浮点计算采用DSP实现，流水线优化后，可每个时钟周期并行处理16个浮点乘法操作，这里分片大小以16为例。由于输入神经元个数并不一定被16整除，所以在每条数据分片进行点积计算时，最后一个分片可能数目不够16，则运算单元将以0填充不足16的部分进行并行乘法计算。

[0094] 二叉浮点加法树3500，将并行浮点乘法3400结构中得到的浮点结果进行累加操作，采用二叉加法树进行并行计算，消除了累加操作的读写依赖，将累加所需的时间复杂度从O(n)将至O(logn)。

[0095] 累加计算3600，由于前向计算处理单元采用分片处理计算，需要将二叉浮点加法树3500计算后得出的结果进行累加，不过累加方式是每隔输出神经元数目进行循环累加操作。

[0096] 激励函数计算3700，采用分段线性近似实现激励函数，计算系数缓存在片上BRAM。

[0097] 数据写回模块3800，配有DMA和FIFO缓存区，数据位宽为32位，负责数据计算结果写回至宿主端内存。

[0098] 该硬件结构支持参数配置，可支持不同规模的神经网络计算。详细配置参数为：

[0099] Data_size:输入神经元数据的规模;

[0100] Input_size:输入神经元的个数,由于预先缓存权值矩阵数据,故这里应该小于片上BRAM所能允许缓存权值参数对应的最大输入神经元个数Max_input;

[0101] Output_size:输出神经元的个数,由于预先缓存权值矩阵数据,故这里应该小于片上BRAM所能允许缓存权值参数对应的最大输出神经元个数Max_output;

[0102] Work_mode:0表示仅进行矩阵乘法计算;1表示进行矩阵乘法和激励函数计算。

[0103] 图10为本发明实施例的现场可编程门阵列平台上进行累加计算的硬件结构示意图3600。结构包括:

[0104] 浮点加法计算3610,由于采用分片思想,需要对点积计算得到的中间值进行累加。中间值数据流是每隔输出神经元的个数N(或后者矩阵的列数)进行累加,累加完毕后再顺序输出。

[0105] 临时值存储BRAM3620,在FPGA内部设置N个存储单元用于存储临时数据,循环将数据流数据累加至对应的BRAM存储单元上,根据输入神经元个数和分片大小的关系判断是否累加结束。由于FPGA内部设计时无法动态的设定用于存储临时值的数量,所以在设计时运算单元设定了支持最大累加数MAX。当输出神经元的个数低于MAX值才能正常进行累加操作。

[0106] 同样对该过程也进行流水线优化,并将启动间隔优化至1个时钟周期,来保证中间值产生和处理的速率保持一致。

[0107] 图11示出了本发明实施例的现场可编程门阵列平台上进行分段线性近似实现激励函数的硬件结构示意图3700。

[0108] 激励函数采用分阶线性近似实现,实现细节如图11所示,与图8不同的是,增加了一条X直接传输到Y的通路,让前向计算运算单元可以仅仅执行矩阵乘法操作而不经过激励函数的处理,这里主要为了实现训练过程中进行误差值计算时所使用的矩阵乘法。由于S型激励函数基本上是关于某点对称,以sigmoid函数为例,sigmoid函数关于(0,0.5)对称,所以当x小于0时,按照1-f(-x)进行计算,这样可以复用硬件逻辑,减少对硬件资源的使用。而且当x等于8时,f(x)等于0.999665,之后便无限接近于1,故当x大于8时,直接对结果赋值为1。

[0109] 图12为本发明实施例的现场可编程门阵列平台上单DMA预存权值参数的前向计算硬件运算单元的计算流程图。

[0110] 首先从DMA依次读取配置数据,根据配置信息读取节点数据。读取节点数据时先将寄存器组a充满后,将flag置0,之后按照flag%2的数值交替输入节点数据值寄存器组a或寄存器组b。同样,根据flag%2的数值读取寄存器组的数据和BRAM缓存的权值数进行并行乘法计算,然后经过二叉加法树求和后进行累加。累加完毕后,根据工作模式选择经过激励函数处理还是直接输出。

[0111] 图13为本发明实施例的现场可编程门阵列平台上双DMA并行读取的前向计算硬件运算单元的结构示意图4000。该硬件结构针对高带宽的FPGA芯片进行前向计算模块设计,采用双DMA并行读取保证高吞吐率。这里分片大小以16为例,结构包括:

[0112] 神经元数据读取模块4100,配有DMA和FIFO缓存区,数据位宽为512位,负责读取输入神经元节点数据,通过移位操作获取16个32位单精度浮点数据。由于数据的传输位宽为

512位，所以要求数据在宿主端内存中要地址对齐。此外对于输入神经元个数不能整除16的情况，需要在宿主端对神经元节点数据矩阵进行充0操作，对每一行的末端填充16-Input_size%16个0，其中Input_size为输入神经元的个数，Input_size%16等于0时无需填充。这里对每个数据复用Output_size次，其中Output_size为输出神经元个数。

[0113] 权值参数数据读取模块4200，配有DMA和FIFO缓存区，数据位宽为512位，负责读取权值参数数据，通过移位操作获取16个32位单精度浮点数据。同样由于数据的传输位宽为512位，所以要求数据在宿主端内存中要地址对齐。此外对于输入神经元个数不能整除16的情况，需要在宿主端对权值参数数据矩阵进行充0操作，在每一列的末尾充16-Input_size%16个0，同样Input_size%16等于0时无需填充。填充完毕后，由于DMA传输需要连续的物理地址，需要将权值参数矩阵的数据存储位置进行调整方便DMA传输。

[0114] 并行浮点乘法4300，将权值参数数据和神经元数据进行并行乘法计算，浮点计算采用DSP实现，流水线优化后，可每个时钟周期并行处理16个浮点乘法操作。

[0115] 二叉浮点加法树4400，将并行浮点乘法4300结构中得到的浮点结果进行累加操作，采用二叉加法树进行并行计算，消除了累加操作的读写依赖，将累加所需的时间复杂度从O(n)将至O(logn)。

[0116] 累加计算4500，由于前向计算处理单元采用分片处理计算，需要将二叉浮点加法树4400计算后得出的结果进行累加，不过累加方式是每隔输出神经元数目进行循环累加操作。该结构和结构3600相同，故不做进一步详细介绍。

[0117] 激励函数计算4600，采用分段线性近似实现激励函数，计算系数缓存在片上BRAM。该结构和结构3700相同，故不做进一步详细介绍。

[0118] 数据写回模块4700，配有DMA和FIFO缓存区，数据位宽为32位，负责数据计算结果写回至宿主端内存。

[0119] 该硬件结构支持参数配置，可支持不同规模的神经网络计算。详细配置参数为：

[0120] Data_size:输入神经元数据的规模；

[0121] Input_size:输入神经元的个数；

[0122] Output_size:输出神经元的个数；

[0123] Work_mode:0表示仅进行矩阵乘法计算;1表示进行矩阵乘法和激励函数计算。

[0124] 图14为本发明实施例的现场可编程门阵列平台上双DMA并行读取的前向计算硬件运算单元的计算流程图。

[0125] 首先从节点DMA读取配置信息，配置运算单元读取节点数据和权值数据的规模以及工作模式。然后，分别从节点DMA和权值DMA读入512位数据，并行移位得到16个神经元节点数据和16个权值参数数据，由于加速器复用节点数据，故每Output_size个时钟周期读取一次节点数据，每1个时钟周期读取一次权值参数数据。数据读取完毕后，依次进行16个并行乘法操作和16输入的二叉加法树求和。将求和结果依次循环加到指定的BRAM存储位置上，并判断是否累加结束。累加结束后，根据工作模式选择直接输出或进行分段近似激励函数处理。

[0126] 图15为本发明实施例的现场可编程门阵列平台上权值更新硬件运算单元的硬件结构示意图5000。采用双DMA并行读取，来保证高吞吐率地计算向量运算。结构包括：

[0127] 向量A数据读取模块5100，配有DMA和FIFO缓冲区，位宽为32位。同时也负责配置参

数的读取。

[0128] 向量B数据读取模块5200，配有DMA和FIFO缓冲区，位宽为32位。

[0129] 计算模块5300，通过不同配置信息进行对应的向量计算。工作模式为0时进行a*A+b*B计算；工作模式为1时进行(a*A+b*B)*B*(1-B)计算。其中a、b为配置参数，A、B分别是读入的向量值。

[0130] 结果写回模块5400，配有DMA和FIFO缓冲区，位宽为32位，将计算结果写回至宿主端内存。

[0131] 该硬件结构支持参数配置，可支持不同规模的向量计算。详细配置参数为：

[0132] Data_size:输入向量数据的规模；

[0133] a:计算所需的系数值；

[0134] b:计算所需的系数值；

[0135] Work_mode:0表示进行a*A+b*B计算；1表示进行(a*A+b*B)*B*(1-B)计算。

[0136] 图16为本发明实施例的现场可编程门阵列平台上权值更新硬件运算单元的计算流程图。

[0137] 首先从DMA A读取配置信息，然后根据配置信息Data_size分别从DMA A和B读取向量的值，并行和配置参数的a和b进行乘法计算后求和，最后根据工作模式选择是否乘以B*(1-B)，将结果通过DMA A写回至宿主端内存。

[0138] 图17为本发明实施例的异构多核可重构计算平台上深度学习加速器的可能一个应用场景及框架示意图。

[0139] 这里应用系统的组成是作为示例说明，本发明并不局限于此。用户对系统发出应用请求时，应用系统控制节点通过调度器将请求分配到对应的计算节点。计算节点在根据具体应用请求将加速任务offload到FPGA进行加速。

[0140] 每个计算节点的整体框架图由硬件层、驱动层、库层、服务层和应用层组成。硬件层是由FPGA、内存和宿主端CPU组成，CPU作为系统的控制器，控制FPGA内部各硬件处理单元(图中简称为DL Module)的运行状态和数据读取，包括前向计算运算单元和权值更新单元。系统计算所需要的权值参数数据和神经元数据仅存储在内存中，通过DMA将数据在内存和硬件处理单元之前传输；驱动层则是根据硬件平台和操作系统编写的硬件驱动；库层则是在驱动基础上封装的应用编程接口API；服务层是面向用户请求提供的深度学习相关计算加速服务；应用层则指深度学习预测算法和训练算法具体的应用，比如说使用卷积神经网络预测算法进行图片分类等等。

[0141] 本领域普通技术人员可以意识到，结合本文中所公开的实施例描述的方法和硬件结构，能够以FPGA和CPU的结合来实现。具体FPGA内部固化IP核的数量和种类看具体应用和FPGA芯片资源限制。专业技术人员可以对每个特定的应用或特定的FPGA芯片来使用不同方式或不同并行度来实现上述所描述的功能，但是这种实现不应认为超出本发明的范围。

[0142] 在本申请所提供的几个实施例中，应该理解到，所揭露的方法和硬件结构，可以通过其它的方式实现。例如，以上所描述深度学习的应用为深度神经网络和卷积神经网络是示意性的。例如，前向计算运算单元中的分片大小以及并行粒度是示意性的，可以根据具体情况进行调整。例如现场可编程门阵列和通用处理器之间的数据传输方式采用AXI总线协议也是示意性。

[0143] 上述实例只为说明本发明的技术构思及特点，其目的在于让熟悉此项技术的人是能够了解本发明的内容并据以实施，并不能以此限制本发明的保护范围。凡根据本发明精神实质所做的等效变换或修饰，都应涵盖在本发明的保护范围之内。

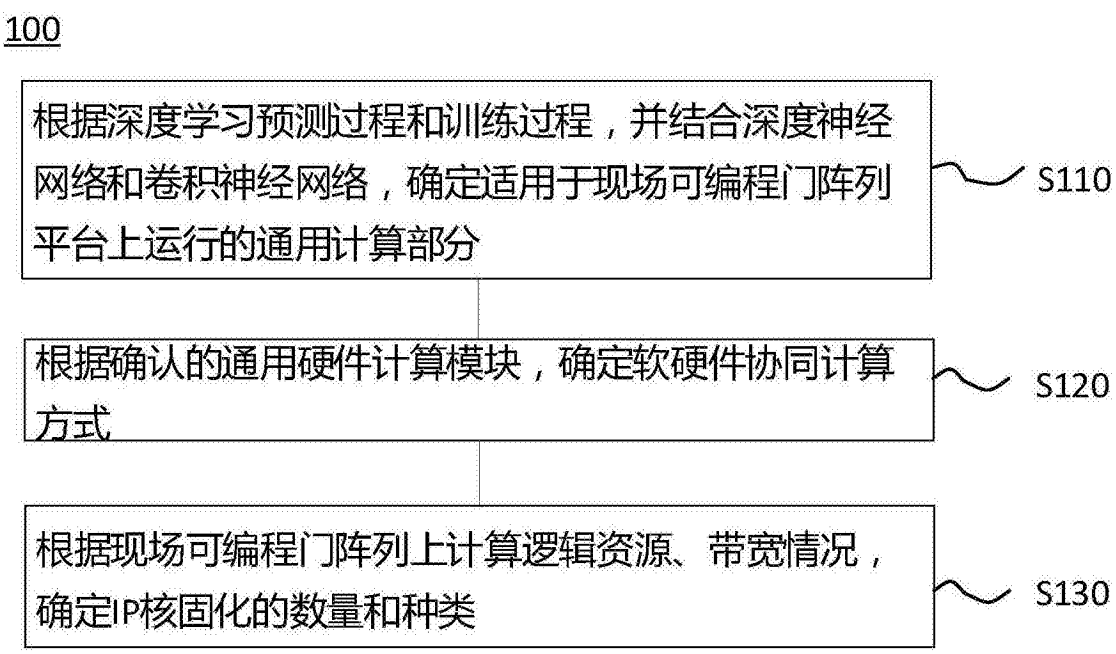

100
根据深度学习预测过程和训练过程，并结合深度神经网络和卷积神经网络，确定适用于现场可编程门阵列平台上运行的通用计算部分
S110
根据确认的通用硬件计算模块，确定软硬件协同计算方式
S120
根据现场可编程门阵列上计算逻辑资源、带宽情况，确定IP核固化的数量和种类
S130

图1

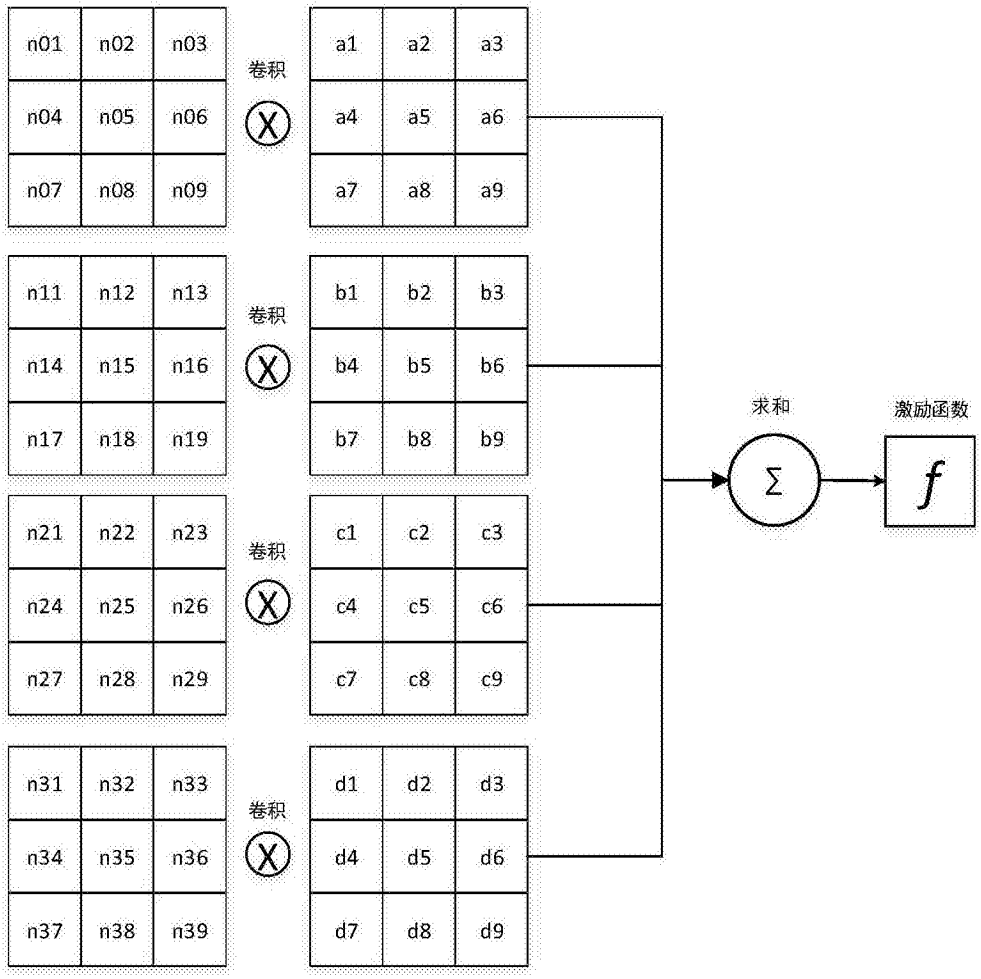

n01 n02 n03
n04 n05 n06
n07 n08 n09
卷积
a1 a2 a3
a4 a5 a6
a7 a8 a9
n11 n12 n13
n14 n15 n16
n17 n18 n19
卷积
b1 b2 b3
b4 b5 b6
b7 b8 b9
n21 n22 n23
n24 n25 n26
n27 n28 n29
卷积
c1 c2 c3
c4 c5 c6
c7 c8 c9
n31 n32 n33
n34 n35 n36
n37 n38 n39
卷积
d1 d2 d3
d4 d5 d6
d7 d8 d9
求和
Σ
激励函数
f

图2

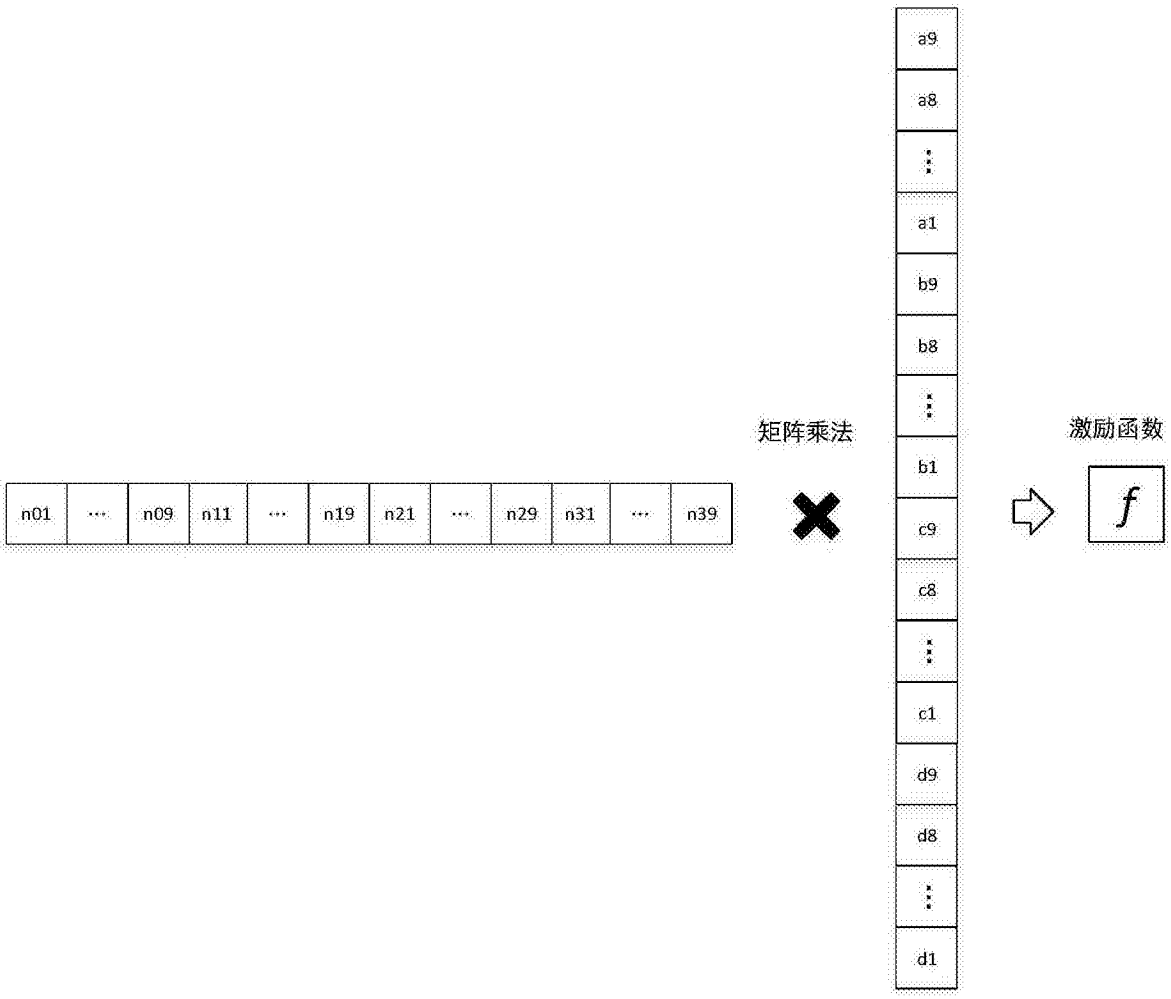

a9
a8
a1
b9
b8
b1
c9
c8
c1
d9
d8
d1
矩阵乘法
激励函数
f
n01
n09
n11
n19
n21
n29
n31
n39

图3

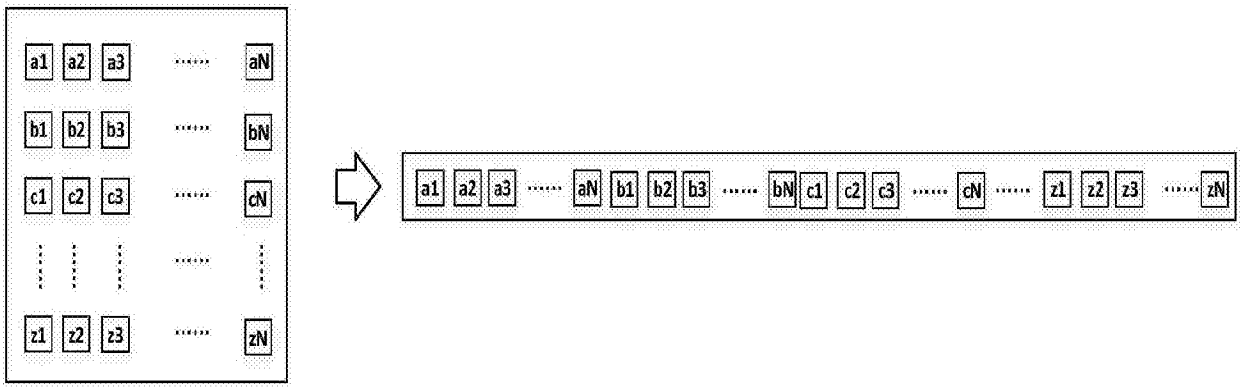

a1 a2 a3 aN
b1 b2 b3 bN
c1 c2 c3 cN
z1 z2 z3 zN
a1 a2 a3 aN b1 b2 b3 bN c1 c2 c3 cN z1 z2 z3 zN

图4

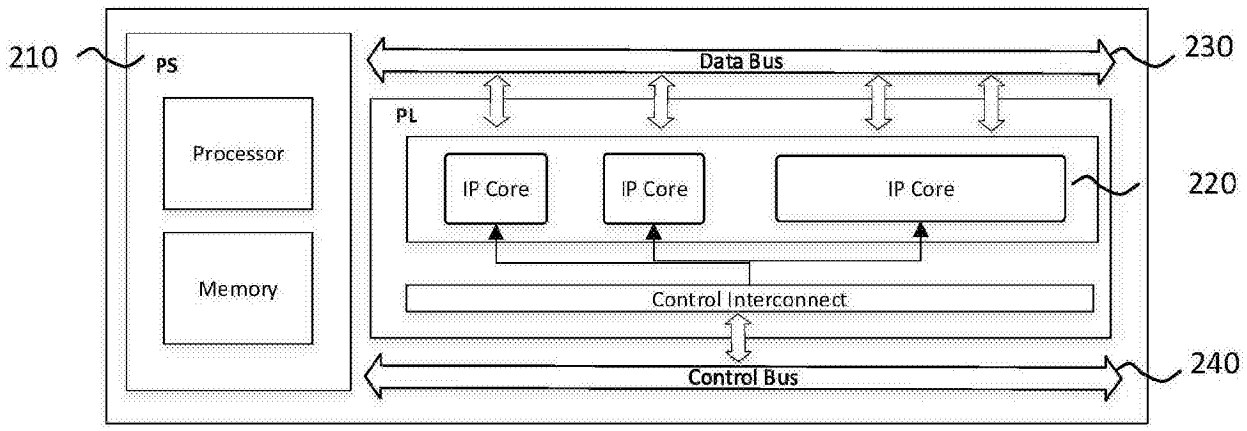

210
PS
Processor
Memory
Data Bus
230
PL
IP Core
IP Core
IP Core
220
Control Interconnect
Control Bus
240

图5

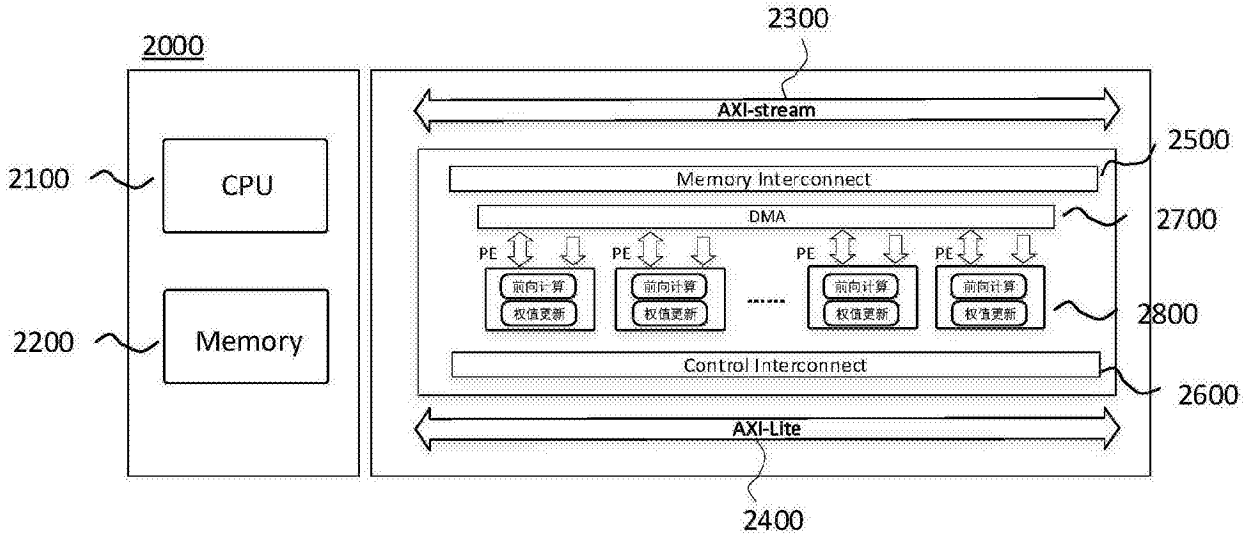

2000
2300
AXI-stream
2100
CPU
2200
Memory
2500
Memory Interconnect
DMA
2700
PE
前向计算
权值更新
2800
Control Interconnect
2600
AXI-Lite
2400

图6

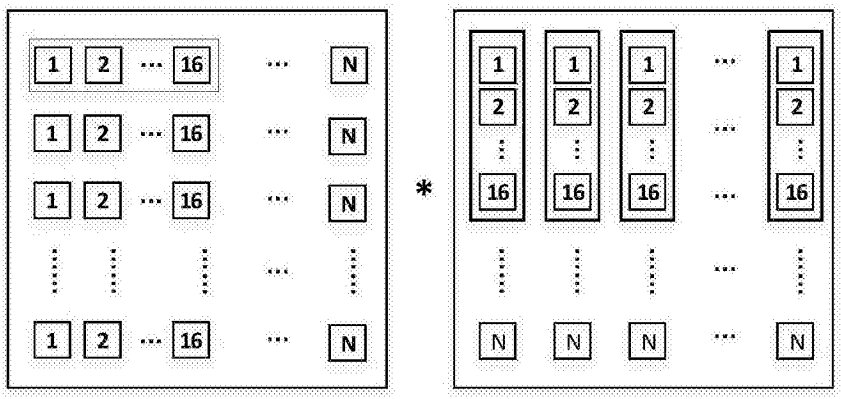

1
2
16
N
*

图7

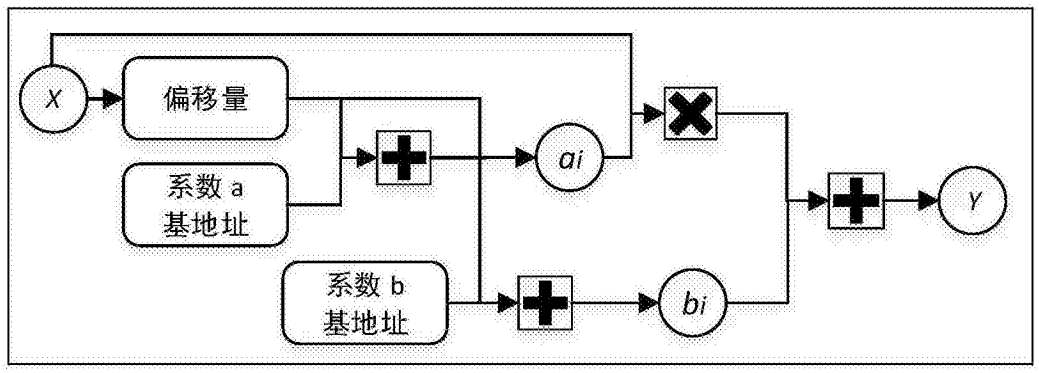

X
偏移量
系数 a
基地址
ai
系数 b
基地址
bi
Y

图8

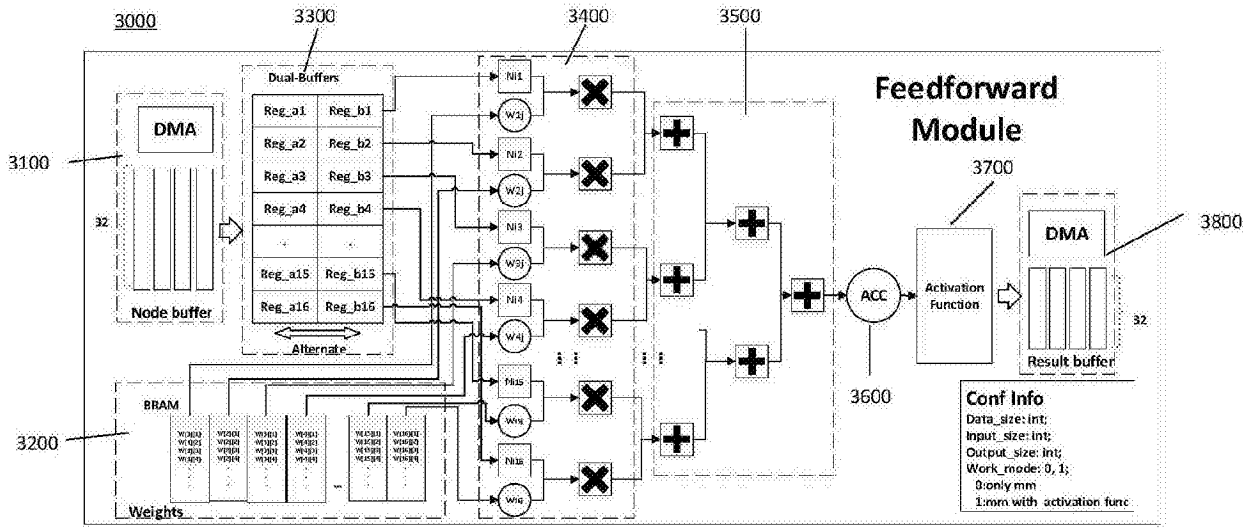

3000
3300
3400
3500
Feedforward
Module
3700
3800
3100
3200
3600
DMA
32
Node buffer
Dual-Buffers
Reg_a1
Reg_b1
Reg_a2
Reg_b2
Reg_a3
Reg_b3
Reg_a4
Reg_b4
Reg_a15
Reg_b15
Reg_a16
Reg_b16
Alternate
BRAM
Weights
ACC
Activation
Function
DMA
32
Result buffer
Conf Info
Data_size: int;
Input_size: int;
Output_size: int;
Work_mode: 0, 1;
0:only mm
1:mm with activation func

图9

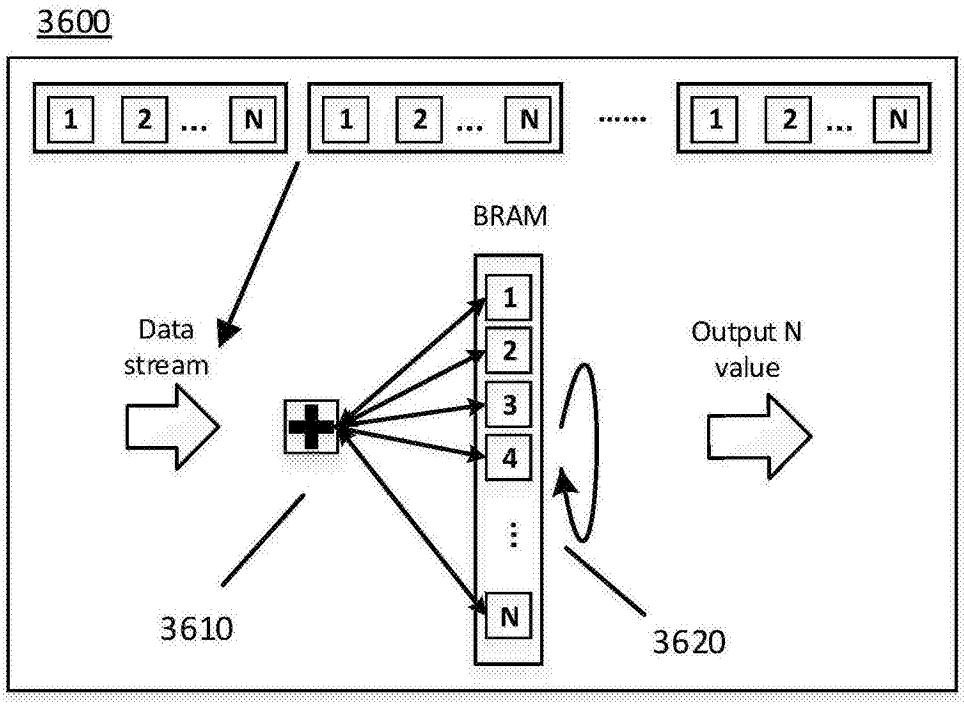

3600
1
2
...
N
.....
BRAM
Data stream
Output N value
3610
3620

图10

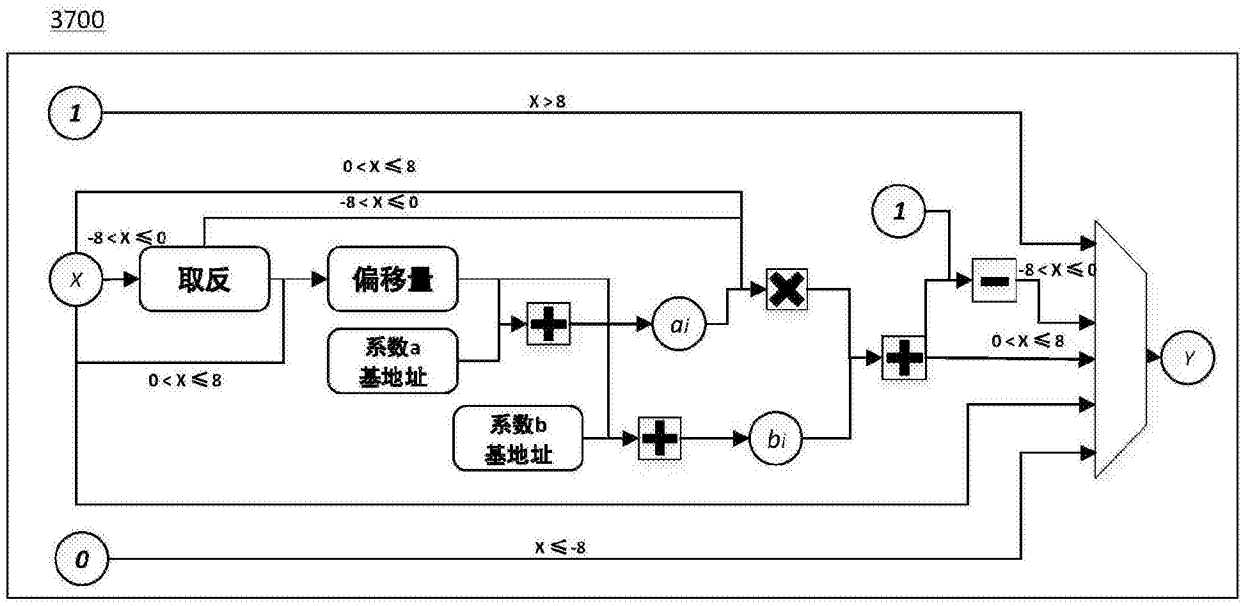

3700
1
X>8
0<X≤8
-8<X≤0
X
取反
偏移量
系数a 基地址
系数b 基地址
ai
bi
0
X≤-8
Y

图11

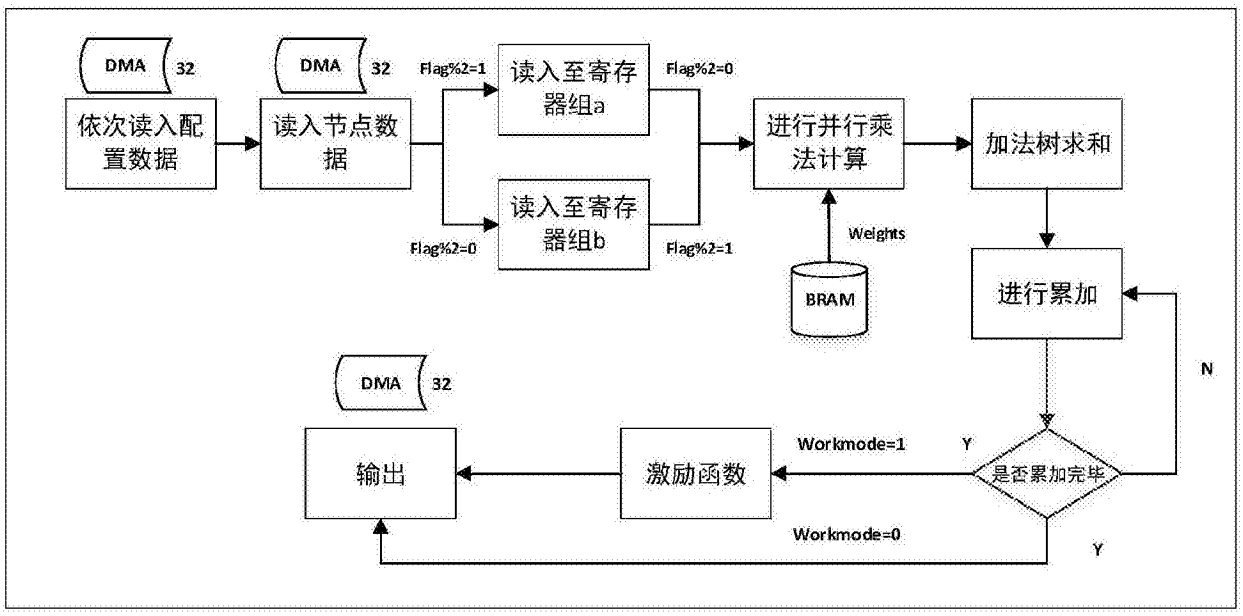

DMA
32
依次读入配置数据
DMA
32
读入节点数据
Flag%2=1
读入至寄存器组a
Flag%2=0
Flag%2=0
读入至寄存器组b
Flag%2=1
进行并行乘法计算
Weights
BRAM
加法树求和
进行累加
N
DMA
32
输出
激励函数
Workmode=1
Y
是否累加完毕
Workmode=0
Y

图12

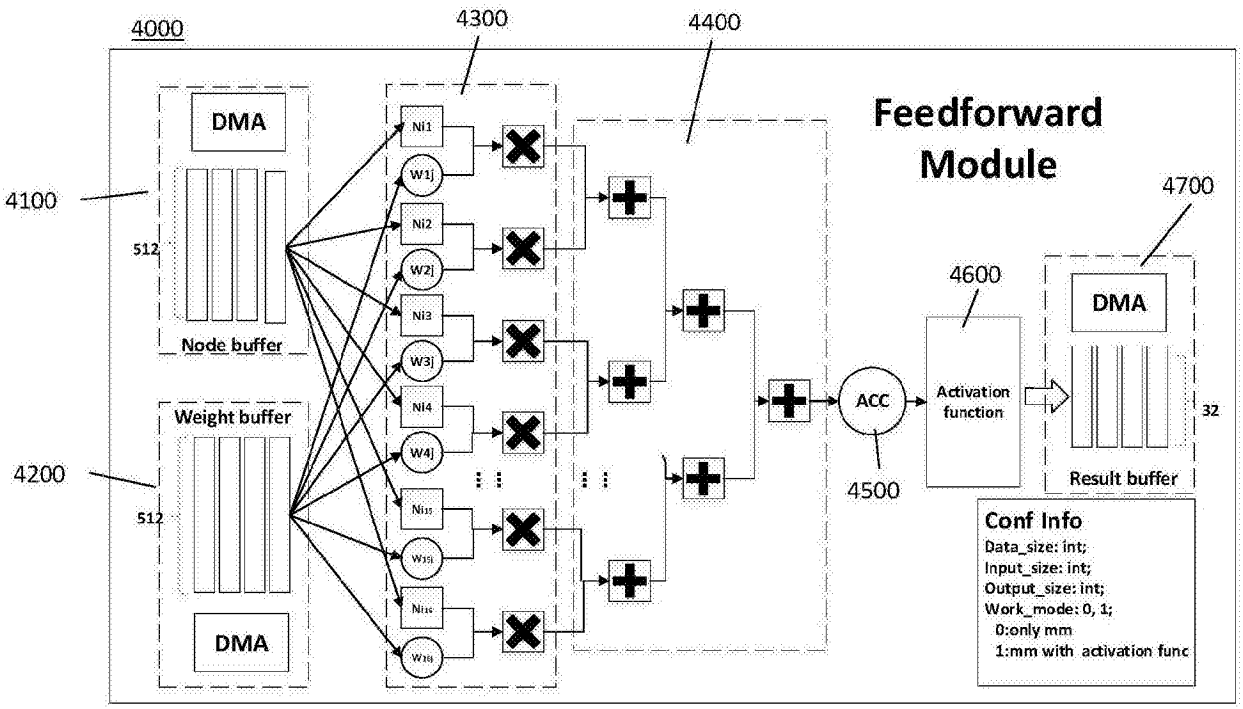

4000
4300
4400
Feedforward Module
4100
DMA
512
Node buffer
4200
Weight buffer
512
DMA
Ni1
W1j
Ni2
W2j
Ni3
W3j
Ni4
W4j
ACC
4500
4600
Activation function
4700
DMA
32
Result buffer
Conf Info
Data_size: int;
Input_size: int;
Output_size: int;
Work_mode: 0, 1;
0:only mm
1:mm with activation func

图13

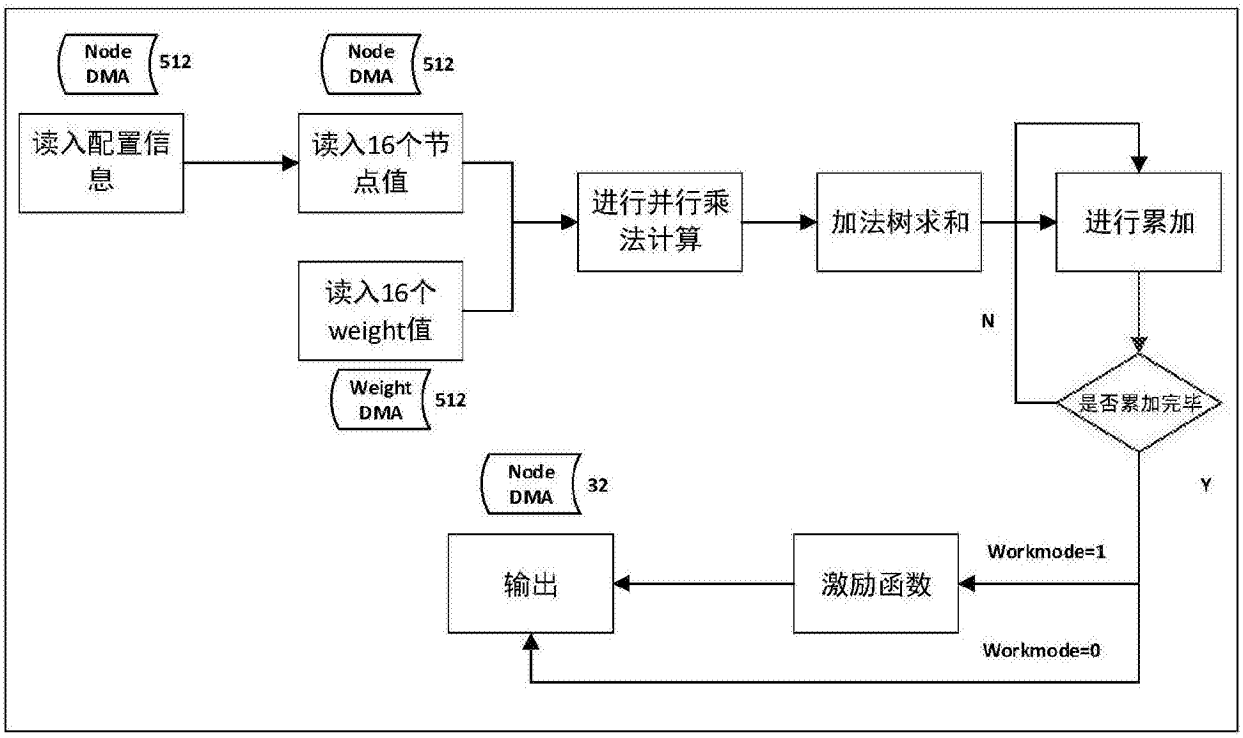

Node DMA
512
Node DMA
512
读入配置信息
读入16个节点值
读入16个weight值
Weight DMA
512
进行并行乘法计算
加法树求和
进行累加
N
是否累加完毕
Y
Node DMA
32
输出
激励函数
Workmode=1
Workmode=0

图14

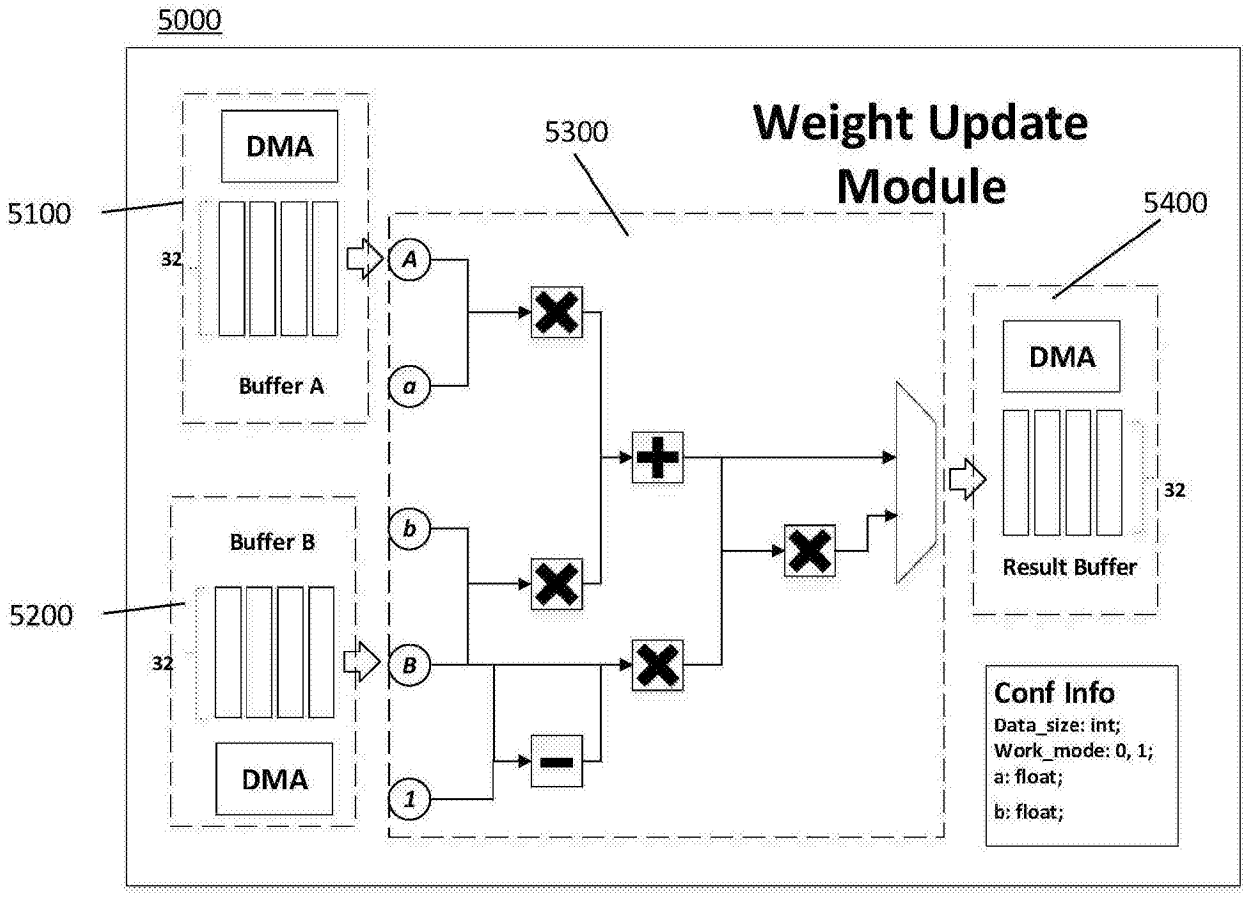

5000
5100
DMA
32
Buffer A
5200
Buffer B
32
DMA
5300
Weight Update Module
A
a
b
B
1
5400
DMA
32
Result Buffer
Conf Info
Data_size: int;
Work_mode: 0, 1;
a: float;
b: float;

图15

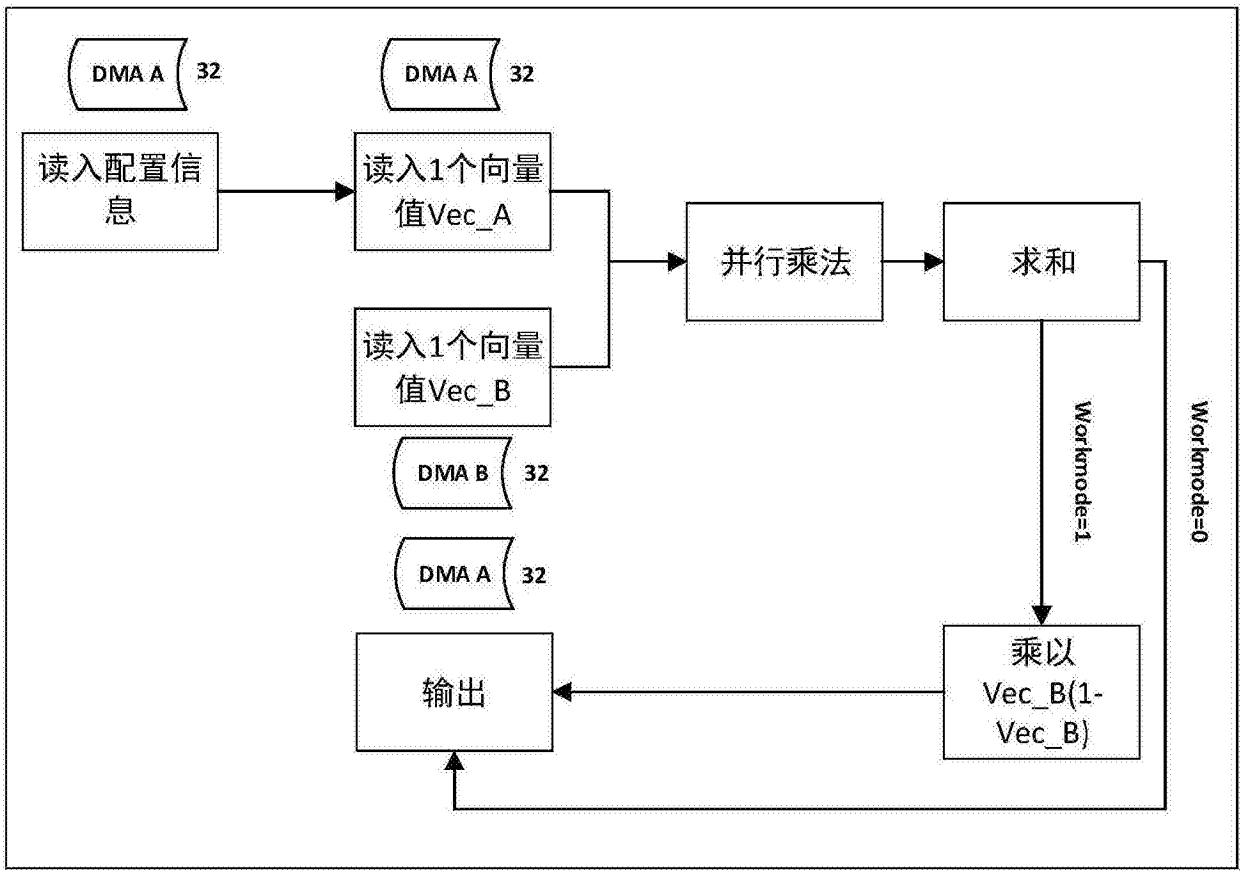

DMA A
32
读入配置信息
DMA A
32
读入1个向量值Vec_A
读入1个向量值Vec_B
DMA B
32
DMA A
32
并行乘法
求和
Workmode=1
Workmode=0
乘以Vec_B(1-Vec_B)
输出

图16

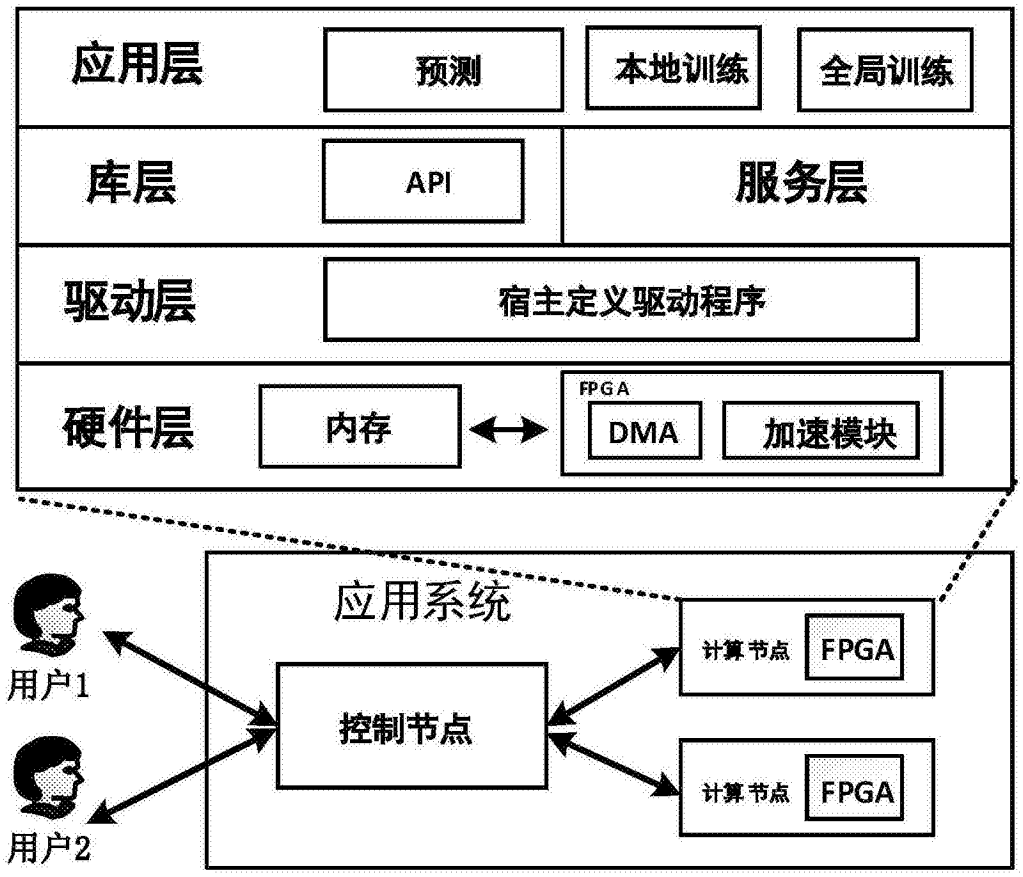

应用层
预测
本地训练
全局训练
库层
API
服务层
驱动层
宿主定义驱动程序
硬件层
内存
FPGA
DMA
加速模块
应用系统
用户1
用户2
控制节点
计算节点
FPGA
计算节点
FPGA

图17