(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 111104767 B
(45)授权公告日 2021.10.01

(21)申请号 201811177869.8
(22)申请日 2018.10.10
(65)同一申请的已公布的文献号
申请公布号 CN 111104767 A
(43)申请公布日 2020.05.05
(73)专利权人 北京大学
地址 100871 北京市海淀区颐和园路5号
(72)发明人 罗国杰 张文泰 姜明
(74)专利代理机构 北京万象新悦知识产权代理有限公司 11360
代理人 黄凤茹
(51)Int.Cl.
G06F 30/327(2020.01)
(56)对比文件
CN 107808364 A,2018.03.16
CN 105630739 A,2016.06.01
US 2017228645 A1,2017.08.10
US 2018046903 A1,2018.02.15
张军阳等.《深度学习相关研究综述》.《计算机应用研究》.2018,
洪启飞.《面向深度学习的FPGA硬件加速平台的研究》.《中国优秀硕士学位论文全文数据库》.2018,
Kaan Kara等.《FPGA-Accelerated Dense Linear Machine Learning: A Precision-Convergence Trade-Off》.《2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)》.2017,
Christopher De Sa等.《Understanding and Optimizing Asynchronous Low-Precision Stochastic Gradient Descent》.《2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)》.2017,
审查员 陈欢

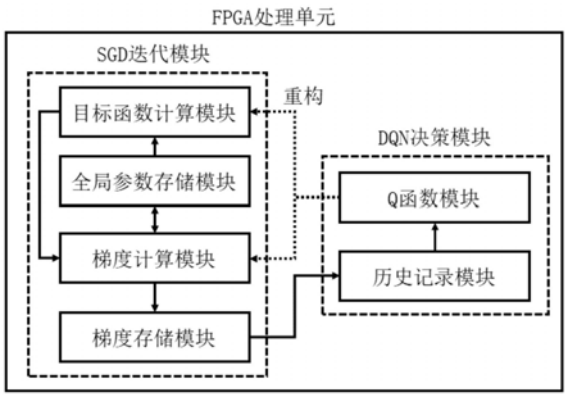
权利要求书2页 说明书7页 附图3页

(54)发明名称

一种针对FPGA的变精度随机梯度下降的结构及设计方法

(57)摘要

本发明公布了一种针对FPGA的变精度随机梯度下降的结构及设计方法,属于计算优化技术领域,是一种在迭代中动态调整精度的随机梯度下降算法(SGD)的动态重构体系结构的新的设计方案,基于动态重构体系结构的方法完成随机梯度下降算法SGD在FPGA上的实现,通过使用深度Q网络DQN对SGD的精度做出预测,由此达到运行时变精度的目的,使得性能更优。本发明通过动态重构方法将SGD迭代模块中的目标函数计算模块和梯度计算模块重新编程,能够使得SGD能够充分利用低精度运算的优势,在保证迭代的收敛性前提下,提高体系结构的计算能力。



CN 111104767 B

1.一种针对FPGA的变精度随机梯度下降结构的设计方法,基于动态重构体系结构的方法完成随机梯度下降算法SGD在FPGA上的实现,通过使用深度Q网络DQN对SGD的精度做出预测,由此达到运行时变精度的目的,使得性能更优;包括:

1)基于一个FPGA处理单元,FPGA处理单元可从外部接受数据,和向外部发送SGD所迭代的参数结果;FPGA处理单元主要包括SGD迭代模块和DQN决策模块;SGD迭代模块包括目标函数计算模块、梯度计算模块、全局参数存储模块、梯度存储模块;DQN决策模块包括Q函数模块,历史记录模块;

2)通过动态重构方法,在切换精度时,利用部分重构将SGD迭代模块中的目标函数计算模块和梯度计算模块重新编程,切换为基于相应精度的版本;收集SGD迭代过程中的信息,并做出决策;由此完成基于FPGA将SGD实现改进为可变精度的SGD实现;

所述方法包括预训练DQN的过程和在FPGA处理单元上实现SGD算法处理对象的过程;具体步骤如下:

A.通过DQN的预训练过程得到训练好的DQN和Q函数的参数;具体执行如下操作:

A1.初始化DQN决策模块的Q函数模块和SGD迭代模块的全局参数存储模块;

A2.当DQN决策模块的Q函数已经收敛,或者已经迭代设定次数,则得到训练好的DQN,保存Q函数的参数到Q函数模块,并转向执行步骤B;

A3.进行一轮SGD的模拟;

参照设定的概率阈值,每次SGD迭代中,DQN决策模块首先基于概率阈值进行一次判断,选择进行预测或选择随机的精度;通过SGD算法计算梯度,并进行一次迭代,保存梯度信息;

A4.将步骤A3的梯度信息和该精度记录到DQN决策模块的历史纪录模块;使用历史纪录模块中的内容作为训练数据,训练DQN;

A5.当SGD算法已经收敛,则重新初始化SGD的全局参数存储模块;转向执行步骤A2;

B.在FPGA处理单元上实现SGD算法处理对象,执行如下操作:

B1.将SGD算法编程到FPGA处理单元上;初始化SGD的全局参数存储模块,开始SGD的迭代过程;

B2.通过SGD迭代模块的目标函数计算模块进行预测计算,并得到目标函数值;如果SGD迭代已经收敛或达到设定迭代次数,则保存结果并结束;由此完成基于FPGA的变精度SGD的实现;

B3.梯度存储模块中的信息即为当前SGD的迭代状态;使用步骤A中训练得到的DQN模型,输入当前SGD的迭代状态,得到预测的精度;

B4.根据得到的精度配置,重构目标函数计算模块和梯度计算模块;

B5.梯度计算模块结合之前目标函数计算模块的结果计算梯度,并配合全局参数存储模块中的参数使用梯度下降法更新全局参数;

继续执行步骤B2。

2.如权利要求1所述针对FPGA的变精度随机梯度下降结构的设计方法,其特征是,步骤B1具体使用工具Xilinx HLS 2018.2将SGD算法编程到FPGA处理单元上。

3.如权利要求1所述针对FPGA的变精度随机梯度下降结构的设计方法,其特征是,采用SGD迭代中的梯度信息作为SGD的迭代状态;具体采用的梯度统计信息包括:梯度,梯度的平方,梯度的方差,一阶动量,二阶动量。

4.如权利要求1所述针对FPGA的变精度随机梯度下降结构的设计方法,其特征是,每轮迭代中,将梯度的统计信息收集到SGD迭代模块的梯度存储模块中;并设置一个历史纪录的长度,使得可以将梯度统计信息组成一个方阵。

5.如权利要求1所述针对FPGA的变精度随机梯度下降结构的设计方法,其特征是,步骤A中得到的训练好的DQN,针对某一类问题的SGD算法可反复使用;某一类问题的SGD算法包括但不限于图像分类或图像重建。

6.如权利要求1～5中任一项所述针对FPGA的变精度随机梯度下降结构的设计方法,其特征是,将该方法应用于包括:深度神经网络的训练,支持向量机和逻辑回归的求解,高精度医学图像的重建。

7.如权利要求6所述针对FPGA的变精度随机梯度下降结构的设计方法,其特征是,将该方法应用于SGD中涉及到大量浮点数运算的情形。

8.一种针对FPGA的变精度随机梯度下降的结构,针对一个FPGA处理单元,FPGA处理单元可从外部接受数据,和向外部发送SGD所迭代的参数结果;其特征是,FPGA处理单元主要包括SGD迭代模块和DQN决策模块;SGD迭代模块包括目标函数计算模块、梯度计算模块、全局参数存储模块、梯度存储模块;DQN决策模块包括Q函数模块,历史记录模块;

SGD迭代模块用于负责调用目标函数计算模块和梯度计算模块来获得必要的信息,用来更新全局参数存储模块中存储的模块;

SGD迭代模块中的梯度存储模块是除普通FPGA上的实现之外增加的模块,用于存储迭代过程中的梯度信息,给DQN决策模块提供输入;存储当前迭代轮的梯度信息,并由梯度计算模块负责更新;

SGD迭代模块的目标函数计算模块用于进行迭代计算SGD算法的目标函数并得到结果;

SGD迭代模块的梯度计算模块用于计算目标函数的梯度,并记录到梯度存储模块中;

SGD迭代模块的全局参数存储模块用于记录全局参数,由SGD迭代模块负责更新;

DQN决策模块使用变精度的SGD,引入动态重构的机制,用于进行预测,根据预测切换精度或选择随机的精度,模拟该精度的迭代,并保存梯度信息;精度配置用于重构目标函数计算模块和梯度计算模块;

DQN决策模块的历史纪录模块用于记录梯度存储模块发送过来的信息,在预训练时记录一定长度的历史信息,用作训练DQN的训练数据;历史纪录模块记录梯度的信息和进行预测的精度记录;历史纪录模块中的内容作为训练数据,用于训练DQN;

DQN决策模块的Q函数模块用于存储DQN中Q函数的参数,并负责根据输入的信息,返回决策。

9.如权利要求8所述针对FPGA的变精度随机梯度下降的结构,其特征是,所述FPGA处理单元采用Xilinx KU115 FPGA芯片。

10.如权利要求9所述针对FPGA的变精度随机梯度下降的结构,其特征是,Xilinx KU115FPGA芯片通过PCIe 3.0接口连接主机/工作站;主机/工作站优选采用Dell Precision T7910Tower Workstation。

一种针对FPGA的变精度随机梯度下降的结构及设计方法

技术领域

[0001] 本发明属于计算优化技术领域,涉及随机梯度下降算法(SGD)的设计方案,具体涉及到一种针对FPGA(Field-Programmable Gate Array,现场可编程门阵列)的变精度随机梯度下降的设计结构及其设计方法。

背景技术

[0002] 随机梯度下降算法(Stochastic Gradient Descent,SGD)被广泛的应用于求解很多最优化问题中。SGD被用来求解最优化问题,使用迭代的方法来或者使目标函数最小的参数。在迭代中,SGD通过标准的梯度下降或者批次梯度下降修正参数。由于涉及到梯度的计算,SGD往往依赖于大量的浮点运算。一个SGD常见的应用领域是机器学习、深度神经网络领域。由于深度神经网络中包含大量的参数,以及非常多的矩阵运算,对计算能力的需求较大,SGD需要一定的优化才能获得优秀的性能。

[0003] 在运算要求很高的应用中,低精度优化是一种常见的加速方法。通过在计算时使用较低精度的浮点数,矩阵运算和SGD可以取得更高的吞吐能力。已经有一些工作探索过低精度SGD,涉及到不同的计算平台。2017年,K.Kara等人提出了一种在FPGA上实现单精度网络的方法,并针对稠密矩阵运算提出了特殊的优化。2017年,斯坦福大学的研究组针对CPU和FPGA,对不同的参数采用不同的精度策略,尤其是大量使用了8比特定点数,取得了十倍左右的性能提升。

[0004] 总的来说,目前SGD的体系结构设计在运行时都采用一致的精度,主要是8比特的整数类型和单精度浮点数。然而,在SGD的迭代过程中,随着“当前解”逐渐逼近“最优解”,梯度的浮动范围越来越窄,可以被使用更少比特的二进制数所表示。已有的很多方法不能在运行时利用这一点,来获取进一步的性能提升。另外,一旦进行精度修改,SGD可能无法保证迭代是否仍然可以收敛,或者深度神经网络是否可以到达有效的精度。

[0005] FPGA是一种通用的可编程器件,其可定制性和可重构性使得其具有很高的灵活性。动态重构技术是一种在用户或者预先设定的控制逻辑下,运行时对体系结构的全部或部分逻辑资源进行功能变换。动态重构技术主要用来应对在运行时处理不同的任务的需求。动态重构可以被划分为两种:全局重构和部分重构。部分重构对芯片的部分逻辑资源重新配置,其余部分不受影响。在FPGA中,特定的部分重构技术可以在运行时重新编程器件。这对于帮助实现SGD的变精度算法有很大的帮助。

发明内容

[0006] 为了克服上述现有技术的不足,本发明提供一种在迭代中动态调整精度的随机梯度下降算法(SGD)的动态重构体系结构的新的设计方案,使得SGD能够充分利用低精度运算的优势,在保证迭代的收敛性前提下,提高体系结构的计算能力。

[0007] 为了配合变精度的需求,本发明引入了强化学习中深度Q网络(DQN)对SGD未来的精度做出预测,既可以保证迭代的准确率,又可以有效地降低精度。DQN将当前的精度和梯

度相关的统计信息作为输入，输出当前状态下的性能和准确度兼顾的精度选择。DQN包含Q函数，用来为DQN决策提供依据。

[0008] 为方便叙述，下面先约定一些记号、术语和变量定义：

[0009] 决策：指当前所采用的精度；

[0010] 状态：指当前SGD的梯度的统计信息，包含之前若干次迭代的历史信息；

[0011] 吞吐量：计算能力，对于浮点数运算，可以被量化为每秒浮点数运算数(Floating Point Operations Per Second，FLOPS)。

[0012] 准确率：当前SGD的目标函数相较于目标的差距，一般可以被量化；在神经网络的训练中，可以由损失函数所推导出；

[0013] 奖励：是一个由吞吐量和准确率构成加权和，吞吐量越高，奖励越大；同时需要保证准确率高于一定水准。

[0014] 本发明提供的技术方案如下：

[0015] 一种针对FPGA的变精度随机梯度下降的设计方法，针对SGD在FPGA上的实现，提出了一种基于动态重构体系结构的方法，使用DQN对随机梯度下降算法SGD精度做出预测，达到运行时变精度的目的，最终可以取得更好的性能。

[0016] 进一步的，本发明所对应的实际应用包含任意的SGD使用。具体地说，可以应用在深度神经网络的训练，支持向量机和逻辑回归的求解，高精度医学图像的重建等等。本发明尤其适合于在SGD中涉及到大量浮点数运算的情形。

[0017] 进一步的，本发明中的设计方案基于一个FPGA处理单元；该FPGA处理单元可以从外部接受数据，向外部发送SGD所迭代的参数结果。FPGA处理单元主要由两个模块构成：SGD迭代模块和DQN决策模块。SGD迭代模块由目标函数计算模块，梯度计算模块，全局参数存储模块，梯度存储模块构成。DQN决策模块由Q函数模块，历史记录模块构成。

[0018] 进一步的，设计方案中引入了动态重构方法。动态重构的目的是为了在切换精度的时候，利用部分重构，将SGD迭代模块中的两个计算模块(目标函数计算模块和梯度计算模块)重新编程，切换为基于相应精度的版本。

[0019] 本发明提供的基于FPGA的变精度SGD的实现方法，利用上述的FPGA处理单元，通过DQN和动态重构，收集SGD迭代过程中的信息，并作出决策，由此将一般的SGD实现改进为可变精度的SGD实现；该实现方法包括如下步骤：

[0020] A.DQN的预训练，具体步骤执行如下：

[0021] A1.初始化DQN决策模块的Q函数模块和SGD迭代模块的全局参数存储模块；

[0022] A2.如果DQN决策模块的Q函数已经收敛，或者已经迭代了合适的次数(一个提前设定的值，比如200轮次的迭代)，则得到训练好的DQN，保存Q函数的参数到Q函数模块，并执行步骤B；

[0023] A3.进行一轮SGD的模拟；在SGD的每次迭代，参照设定的概率阈值，DQN决策模块首先基于概率阈值进行一次判断，选择是进行预测或者选择随机的精度；举例来说，概率阈值如果是0.3，也就是有70％的可能选择DQN决策模块预测的结果，30％的可能选择随机的精度；DQN决策模块在进行预测时，从梯度存储模块中获取必要的信息，并输出所预测的精度；

[0024] A4.SGD算法计算梯度，并进行一次迭代，然后保存梯度信息，并将梯度的信息和该精度记录到DQN决策模块的历史纪录模块，使用历史纪录模块中的内容作为训练数据，训练

DQN；

[0025] A5.如果SGD算法已经收敛了，重新初始化SGD的全局参数存储模块；执行步骤A2。

[0026] 通过上述DQN的预训练过程得到训练好的DQN和Q函数的参数。

[0027] B.在FPGA处理单元上实现SGD算法处理对象，执行如下操作：

[0028] B1.使用工具(如Xilinx HLS 2018.2)将SGD算法编程到FPGA处理单元上；初始化SGD的全局参数存储模块，开始SGD的迭代过程；

[0029] B2.通过SGD迭代模块的目标函数模块进行计算，如果SGD迭代已经收敛，或者达到合适的次数(是一个提前设定的值，比如200轮次的迭代)，则保存结果并结束；

[0030] B3.使用步骤A中训练得到的模型DQN，输入当前SGD的迭代状态(梯度存储模块中的信息)，得到预测的精度；

[0031] B4.根据得到的精度配置，重构目标函数计算模块和梯度计算模块；

[0032] B5.梯度计算模块结合之前目标函数模块的结果计算梯度，并配合全局参数存储模块中的参数使用梯度下降法更新全局参数；执行步骤B2。

[0033] 步骤A中得到的DQN，对于针对某一类问题的SGD算法(比如图像分类或者重建)，可以反复使用。

[0034] 通过上述步骤，完成基于FPGA的变精度SGD的实现。

[0035] 本发明还提供一种针对FPGA的变精度随机梯度下降的结构，针对一个FPGA处理单元，该FPGA处理单元可以从外部接受数据，向外部发送SGD所迭代的参数结果；FPGA处理单元主要包括两个模块：SGD迭代模块和DQN决策模块；SGD迭代模块包括：目标函数计算模块、梯度计算模块、全局参数存储模块和梯度存储模块；DQN决策模块由Q函数模块和历史记录模块构成。

[0036] SGD迭代模块除一般的FPGA上的实现之外，增加一个梯度存储模块；SGD迭代模块还用于负责调用目标函数模块和梯度计算模块来获得必要的信息，用来更新全局参数存储模块中存储的模块；

[0037] SGD迭代模块的梯度存储模块用于存储迭代过程中的梯度信息，给DQN决策模块提供输入；存储当前迭代轮的梯度信息，由梯度计算模块负责更新；

[0038] SGD迭代模块的目标函数模块用于进行迭代计算SGD算法的目标函数并得到结果；

[0039] SGD迭代模块的梯度计算模块用于计算目标函数的梯度，并记录到梯度存储模块中；

[0040] SGD迭代模块的全局参数存储模块用于记录全局参数，由SGD迭代模块负责更新；

[0041] DQN决策模块用于进行预测，根据预测切换精度或选择随机的精度，模拟该精度的迭代，然后保存梯度信息；精度配置用于重构目标函数计算模块和梯度计算模块；DQN决策模块使用了变精度的SGD，通过SGD迭代模块的信息，做出保证SGD准确率的精度决策；通过引入动态重构的机制，在运行时就可以完成精度模块的重新编程；

[0042] DQN决策模块的历史纪录模块用于记录梯度存储模块发送过来的信息，在预训练时记录一定长度的历史信息，用作训练DQN的训练数据；历史纪录模块记录梯度的信息和进行预测的精度记录；历史纪录模块中的内容作为训练数据，用于训练DQN；

[0043] Q函数模块用于存储DQN中Q函数的参数，并负责根据输入的信息，返回决策。

[0044] 本发明的有益效果：

[0045] 本发明提供一种针对FPGA的变精度随机梯度下降的设计方法及结构，实现在迭代中动态调整精度的随机梯度下降算法(SGD)的动态重构体系结构，使得SGD能够充分利用低精度运算的优势，在保证迭代的收敛性前提下，提高体系结构的计算能力。利用本发明提供的技术方案，通过在SGD的硬件实现中引入动态重构，使得SGD可以充分利用变精度带来的吞吐量优势，取得更高的性能。

附图说明

[0046] 图1本发明中FPGA处理单元的设计方案框图。

[0047] 图2本发明中预训练DQN的步骤流程框图。

[0048] 图3本发明中在FPGA处理单元上执行SGD算法的步骤流程框图。

具体实施方式

[0049] 下面结合附图，通过实施例进一步描述本发明，但不以任何方式限制本发明的范围。

[0050] 本发明的具体实施方式如下：

[0051] 本发明的设计方案主要由FPGA处理单元构成。FPGA处理单元主要由两个模块构成：SGD迭代模块和DQN决策模块；SGD迭代模块由目标函数计算模块，梯度计算模块，全局参数存储模块，梯度存储模块构成；DQN决策模块由Q函数模块，历史记录模块构成；具体地：

[0052] SGD迭代模块除一般的FPGA上的实现外，增加一个梯度存储模块，用来存储迭代过程中的梯度信息，给DQN决策模块提供输入；同时，也负责调用目标函数模块和梯度计算模块来获得必要的信息，用来更新全局参数存储模块中存储的参数；

[0053] 目标函数计算模块：目标函数负责计算SGD算法的目标函数的值；

[0054] 梯度计算模块：该模块负责计算目标函数的梯度，并记录到梯度存储模块中；

[0055] 全局参数存储模块：负责记录全局参数，由SGD迭代模块负责更新；

[0056] 梯度存储模块：存储当前迭代轮的梯度信息，由梯度计算模块负责更新；

[0057] DQN决策模块：通过SGD迭代模块的信息，做出保证SGD准确率的精度决策；

[0058] Q函数模块：存储了DQN中Q函数的参数，并负责根据输入的信息，返回决策；

[0059] 历史记录模块：记录梯度存储模块发送过来的信息，在预训练时记录一定长度的历史信息，用作训练DQN的训练数据。

[0060] 和已有的一般方案不同，本发明中使用了变精度的SGD实现。一般的低精度实现中，对SGD迭代的准确率并没有进行保证。在本发明的实现方法中，通过DQN保证了不会让精度下降到损失准确率的程度。通过引入了动态重构的机制，在运行时就可以完成精度模块的重新编程。

[0061] 本发明中，DQN涉及到了三个概念：决策，状态和奖励。一般地说，决策就是DQN返回给SGD的精度。SGD中一般会包含非常多的变量，举例来说，在深度神经网络的训练中，至少可以将变量划分为权重、激活值(Activations)和梯度值三类。不失一般性，本发明的具体实施例中不对这些变量做更细的划分，对所有的变量应用同样的精度配置；用户可以根据自身的需求做进一步的细化。

[0062] 状态是SGD迭代过程中的各种信息的历史。本发明中采用SGD迭代中的梯度信息作

为状态的基本组成。具体地，我们采用了如下的梯度统计信息：梯度，梯度的平方，梯度的方差，一阶动量，二阶动量。每轮迭代中，梯度的统计信息被收集到SGD迭代模块的梯度存储模块中。一般会设置一个历史纪录的长度，本发明中记录了5次迭代的历史，这样正好可以将梯度统计信息组成一个方阵。

[0063] 奖励是DQN中Q函数的出书，代表某个状态下做出特定决策可以获得的收益。本发明中，对于SGD的关注有两点：计算模块的吞吐量和迭代的准确率。因此，本发明的实现方法中将奖励设置为以吞吐量和准确率为输入的估价函数的加权和。权重可以根据用户的实际情况调整。为了保证准确率，准确率的估价函数可以设置成在低于某个阈值的时候有较大的惩罚。举例来说，在图像分类的神经网络训练中，如果希望准确率不要低于90％，那么可以设置估价函数为y＝log(x-90％)，并加大准确率的权重。

[0064] 图1描述了FPGA处理单元的设计方案框图。整个框图主要由两个部分构成：SGD迭代模块和DQN决策模块。SGD迭代模块由目标函数计算模块，梯度计算模块，全局参数存储模块，梯度存储模块构成；DQN决策模块由Q函数模块，历史记录模块构成。在运算中，目标函数需要从全局参数存储模块读取参数计算目标函数；目标函数的结果被用来计算梯度；梯度计算模块使用目标函数的值和全局参数计算梯度，并更新全局参数，同时将梯度的各种信息存储在梯度存储模块之中；Q函数模块使用DQN对精度做出决策，并重构两个计算模块。在预训练中，还多两个连接：梯度存储模块将信息发送给历史记录模块；历史记录模块使用这些信息对Q函数进行训练。

[0065] 图2描述了预训练DQN的流程框图，具体的执行步骤如下：

[0066] 1.将SGD和DQN都实现在FPGA处理单元上；初始化DQN的Q函数模块和SGD的全局参数存储模块；

[0067] 2.判断DQN是否已经收敛，如果是，则跳转步骤7；

[0068] 3.按照一定的概率阈值ε，DQN可以选择使用自身的Q函数和输入的梯度信息预测一个精度，或者选择随机的精度；

[0069] 4.SGD使用DQN返回的精度，进行一个轮次的迭代，整理得到梯度信息和准确度，并计算出奖励，保存到历史纪录模块中；

[0070] 5.DQN使用历史纪录模块中的信息，更新Q函数；

[0071] 6.判断SGD是否已经收敛，如果是则重新初始化SGD的全局参数存储模块；执行步骤2。

[0072] 7.保存DQN的参数，结束过程。

[0073] 图3描述了在FPGA处理单元上执行SGD算法的步骤流程框图，具体的执行步骤如下：

[0074] 1.将SGD和DQN都实现在FPGA处理单元上；初始化SGD的全局参数存储模块；

[0075] 2.SGD的目标函数模块采用SGD算法进行计算，并得到目标函数值；

[0076] 3.判断SGD是否已经收敛，如果是，则跳转步骤7；

[0077] 4.向DQN决策模块发送梯度存储模块中的信息，得到返回的精度；

[0078] 5.使用DQN返回的精度，重构目标函数计算模块和梯度计算模块；

[0079] 6.梯度计算模块结合之前目标函数模块的结果计算梯度，并配合全局参数存储模块中的参数更新参数；执行步骤2；

[0080] 7.保存DQN的参数,结束过程。

[0081] 下面通过实例对本发明做进一步说明。

[0082] 实施例:

[0083] 本实施例在MNIST(Modified National Institute of Standards and Technology)数据集([LeCun et al.,1998a]Y.LeCun,L.Bottou,Y.Bengio,and P.Haffner."Gradient-based learning applied to document recognition." Proceedings of the IEEE,86(11):2278-2324,November 1998.)上训练一个LeNet-5模型(来源和MNIST一致),使用标准的SGD算法训练,FPGA处理单元采用Xilinx KU115FPGA芯片。本实施例使用一台工作站(不限,比如Dell Precision T7910 Tower Workstation)和FPGA芯片进行通信,传输训练数据集。在本例中,我们要求准确率不低于90%,因此将准确率的估价函数设置为y=log(x-90%)。本实施例采用全局的双精度(FP32)和单精度(FP16)浮点数,8比特(8bit)定点数来表示LeNet-5模型的参数,将这些不同的精度作为DQN的决策选择。在本实施例中,FPGA芯片(Xilinx KU115 FPGA芯片)和主机(工作站)通过PCIe 3.0接口连接。训练数据集一开始存储在主机的主存中。使用Xilinx Vivado 2018.2工具,综合Xilinx HLS 2018.2中得到的IP(Intellectual Property)核和Verilog代码。在Xilinx Vivado 2018.2工具中,使用部分重构(Partial Reconfiguration)功能,设置好可以重构的查找表、寄存器、触发器和块内存。主要是将神经网络中矩阵运算和梯度计算的部分标记为可重构的部分。具体的执行步骤如下:

[0084] A.训练DQN;

[0085] A1.使用高层次综合工具,比如Xilinx Vivado® High-Level Synthesis 2018.2(Xilinx HLS 2018.2),参照图1,对SGD算法进行不同精度的实现,预先得到这些不同精度下的FPGA设计的吞吐量和资源消耗。

[0086] A2.开始在FPGA处理单元进行DQN的预训练。参考图2的步骤,训练DQN。DQN的状态即为当前的梯度统计信息和精度设定。用步骤A1中得到的吞吐量和LeNet-5模型的精度作为奖励。训练DQN时,以DQN的预测准确率作为标准,设定一个阈值并在达到阈值时结束迭代。如果训练时间太长,可以设定一个迭代次数的硬上限,如10000次,在迭代次数达到之后结束迭代。

[0087] B.使用步骤A中训练好的DQN,参照图1设计FPGA处理单元的体系结构。然后参照图3执行,具体步骤如下:

[0088] B1.将训练好的DQN和LeNet-5按照图3所示的设计,用高层次综合工具或者直接使用Verilog语言编程,得到一个可以编程FPGA的比特流文件(Bitstream File),烧制到KU115芯片上。

[0089] B3.此时,使用主机和KU115的PCIe连接,从主机发送LeNet-5模型训练所需要的数据,参照图3开始训练。

[0090] 本实施例中一共使用了三种精度设置:全局的双精度(FP32)和单精度(FP16)浮点数,8比特(8bit)定点数。在测试中,我们发现使用本发明实现方法之后,会在迭代进行到一定阶段的时候将精度切换为FP16,接着进一步降低为8比特。在SGD算法的2000个迭代轮次中,三种精度大约各占有了1/3的轮次。本发明首次在SGD迭代中引入了动态的精度,相比之前的方法,能够更好的利用硬件的计算单元,尤其是针对FPGA。

[0091] 本发明同时也测试了全过程使用FP32的SGD算法，在一个比较测试中和采用了本发明实现方法的例子进行了比较。基于本发明的SGD算法，相比于使用全局FP32的版本，在吞吐量上要增加了约330％，总的迭代时间缩短了约75％。

[0092] 综上，本发明提供一种在迭代中动态调整精度的SGD的动态重构体系结构的新的实现方法，适用于SGD迭代中存在大量浮点数算术运算的场景。本发明引入动态精度，使得SGD能够充分利用低精度运算的优势，并且保证迭代的收敛性；另外在体系结构中引入动态重构，最终提高体系结构的计算能力。

[0093] 需要注意的是，公布实施例的目的在于帮助进一步理解本发明，但是本领域的技术人员可以理解：在不脱离本发明及所附权利要求的精神和范围内，各种替换和修改都是可能的。因此，本发明不应局限于实施例所公开的内容，本发明要求保护的范围以权利要求书界定的范围为准。

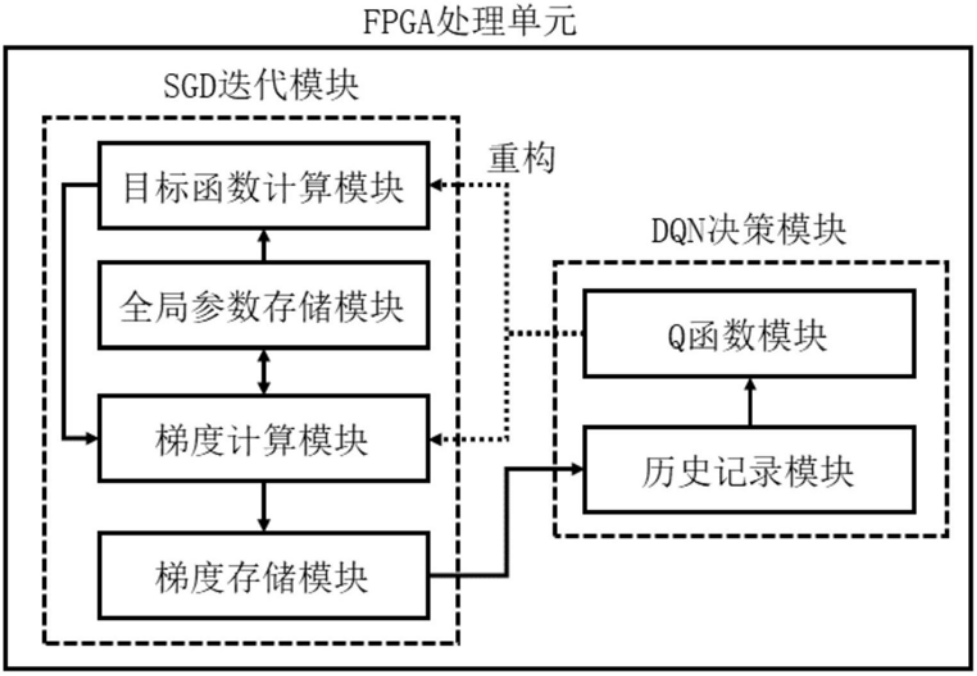

FPGA处理单元
SGD迭代模块
目标函数计算模块
全局参数存储模块
梯度计算模块
梯度存储模块
重构
DQN决策模块
Q函数模块
历史记录模块

图1

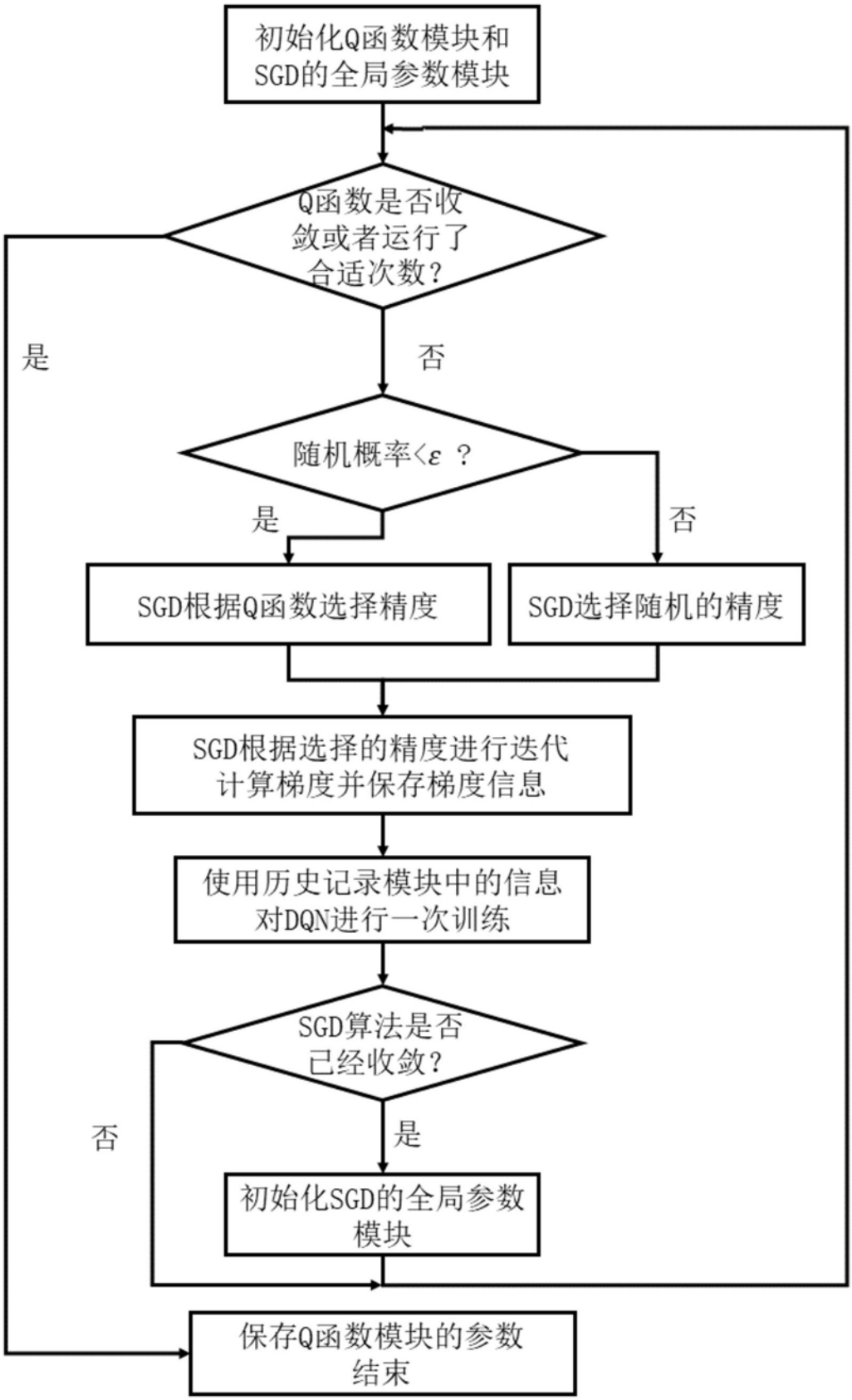

初始化Q函数模块和SGD的全局参数模块
Q函数是否收敛或者运行了合适次数？
是
否
随机概率<ε ？
是
否
SGD根据Q函数选择精度
SGD选择随机的精度
SGD根据选择的精度进行迭代计算梯度并保存梯度信息
使用历史记录模块中的信息对DQN进行一次训练
SGD算法是否已经收敛？
否
是
初始化SGD的全局参数模块
保存Q函数模块的参数结束

图2

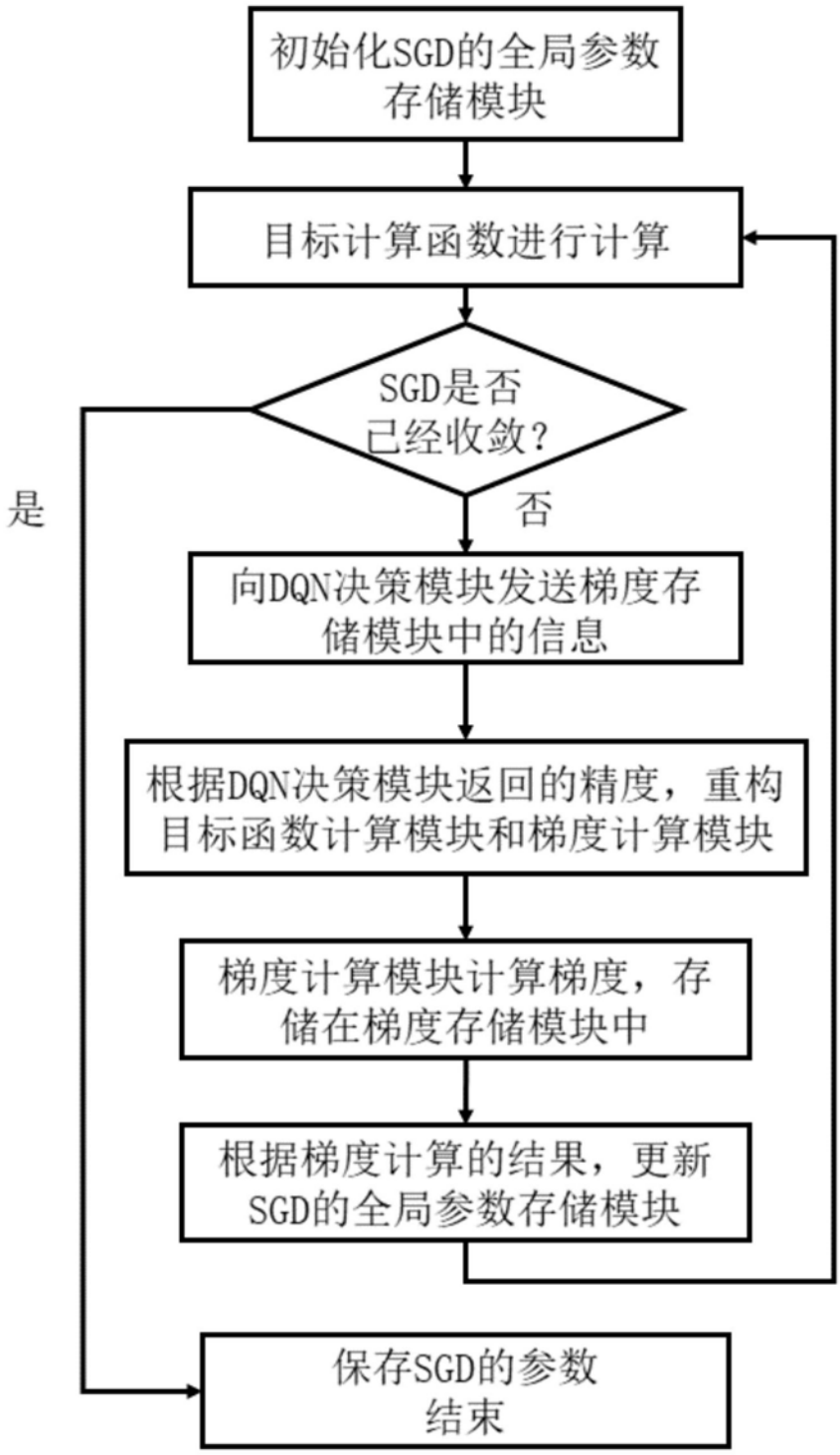

初始化SGD的全局参数存储模块
目标计算函数进行计算
SGD是否已经收敛？
是
否
向DQN决策模块发送梯度存储模块中的信息
根据DQN决策模块返回的精度，重构目标函数计算模块和梯度计算模块
梯度计算模块计算梯度，存储在梯度存储模块中
根据梯度计算的结果，更新SGD的全局参数存储模块
保存SGD的参数
结束

图3