(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 106650922 B
(45)授权公告日 2019.05.03

(21)申请号 201610865581.4

(22)申请日 2016.09.29

(65)同一申请的已公布的文献号
申请公布号 CN 106650922 A

(43)申请公布日 2017.05.10

(73)专利权人 清华大学
地址 100084 北京市海淀区清华大学

(72)发明人 张悠慧 季宇

(74)专利代理机构 北京睿邦知识产权代理事务所(普通合伙) 11481
代理人 张丽新

(51)Int.Cl.
G06N 3/063(2006.01)

(56)对比文件
CN 101310294 A,2008.11.19,
CN 103930908 A,2014.07.16,
US 2014337663 A1,2014.11.13,
CN 105719000 A,2016.06.29,
全钢 等."基于VHDL的神径网络模型库的建立与实现".《微计算机信息》.2002,第18卷(第7期),

审查员 张丽娜

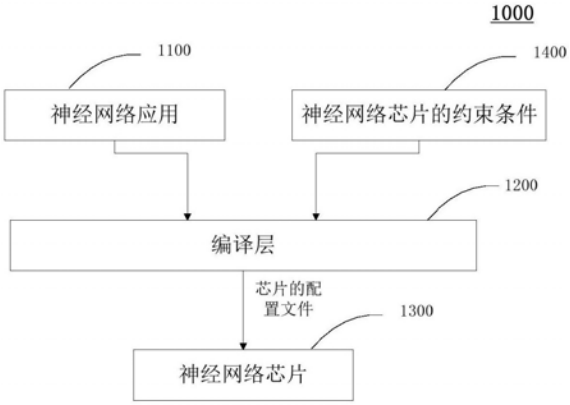
权利要求书6页 说明书23页 附图6页

(54)发明名称

硬件神经网络转换方法、计算装置、软硬件协作系统

(57)摘要

将神经网络应用转换为满足硬件约束条件的硬件神经网络的硬件神经网络转换方法、计算装置、编译方法、神经网络软硬件协作系统,该方法包括:获得神经网络应用对应的神经网络连接图;将神经网络连接图拆分为神经网络基本单元;将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络;将得到的基本单元硬件网络按照拆分的顺序连接起来,生成硬件神经网络的参数文件。本发明提出了一种全新的神经网络和类脑计算的软硬件体系,在神经网络应用和神经网络芯片之间加上了一个中间编译层,解决了神经网络应用与神经网络应用芯片之间的适配问题,同时解耦合了应用和芯片的开发。



CN 106650922 B

1.一种将神经网络应用转换为满足硬件约束条件的硬件神经网络的硬件神经网络转换方法,包括:

神经网络连接图获得步骤,获得神经网络应用对应的神经网络连接图,神经网络连接图是一个有向图,图中的每个节点表示一层神经元,每条边表示层间的连接关系;

神经网络连接图拆分步骤,将神经网络连接图拆分为神经网络基本单元,每个神经网络基本单元中,只有入节点和出节点,不存在中间层节点,入节点和出节点之间全相联,而且入节点中的神经元的所有出度在该基本单元内,出节点中的每个神经元的所有入度在该基本单元内;

神经网络基本单元转换步骤,将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络,称之为基本单元硬件网络,一个神经网络基本单元对应于一个或多个神经网络硬件的基本模块虚拟体,每个神经网络硬件的基本模块虚拟体均满足神经网络硬件的基本模块的连接度约束条件,且能够直接映射到神经网络硬件的基本模块;

基本单元硬件网络连接步骤,将得到的基本单元硬件网络按照拆分的顺序连接起来,生成硬件神经网络的参数文件。

2.根据权利要求1所述的硬件神经网络转换方法,还包括,在神经网络应用具有卷积层的情况下,在神经网络连接图拆分步骤之前,对于神经网络应用的卷积层进行网络压缩,包括:

获得每一卷积层的多个特征图;

利用DPP提取多样性子集的方法,将这些特征图在所有样本上产生的输出之间的相似性作为DPP算法相关联的矩阵元,利用DPP得到多样性最高的子集,保留该子集,丢弃掉其他特征图节点,将丢弃的特征图所对应的向量投影到保留的特征图所张成的线性空间中,用丢弃的特征图的投影长度与其原向量长度的比值作为加权系数,将丢弃的特征图与下一层神经元的连接权重加权累加到保留的特征图与下一层神经元的连接权重上。

3.根据权利要求1所述的硬件神经网络转换方法,所述神经网络基本单元转换步骤包括:

对每个神经网络基本单元重建网络拓扑;以及

针对重建的网络拓扑,进行权重参数确定。

4.根据权利要求3所述的硬件神经网络转换方法,重建网络拓扑包括完全展开操作,经过完全展开,神经网络基本单元被分解为了基本模块虚拟体之间的相互连接,所述完全展开操作包括:

在神经网络基本单元相关联的第一规模的矩阵乘法和/或卷积的大矩阵操作超过了神经网络硬件的基本模块支持的第二规模的小矩阵操作的情况下,执行下述操作:

将第一规模的大矩阵操作拆分为第三数目个第二规模的小矩阵操作,每个小矩阵操作由一个基本模块虚拟体完成;

将针对第一规模的大矩阵操作的输入数据分解为第三数目份,并传送给该第三数目个第二规模的小矩阵操作,此为多播操作;

将来自第三数目个第二规模的小矩阵操作的运算结果汇总为等价于第一规模的大矩阵操作的运算结果,此为归约操作,

在神经网络硬件芯片具有支持多播操作的第一额外模块的情况下，将多播操作分配为由所述第一额外模块虚拟体来执行，否则由多播操作由第一组基本模块虚拟体来完成；

在神经网络硬件芯片具有支持归约操作的第二额外模块的情况下，将归约操作分配为由所述第二额外模块虚拟体来执行，否则由多播操作由第二组基本模块虚拟体来完成。

5.根据权利要求4所述的硬件神经网络转换方法，在神经网络硬件芯片上基本模块数额不足的情况下，采用时分方法来利用基本模块。

6.根据权利要求4所述的硬件神经网络转换方法，重建网络拓扑还包括在完全展开操作之前进行重编码操作，包括：

利用自编码器进行层间数据重编码，自编码器是神经网络，由3层神经元组成，包括输入层、隐藏层和输出层，其中输出层的节点数和输入层节点数相同，隐藏层的节点数量多于层间向量数据的维度，训练该网络，使得输出层的值与输入层的值尽可能相近，其中输入层和输出层的精度为神经网络应用的精度，隐藏层采用神经网络硬件基本模块之间的传输数据的精度，自编码器被转换为编码器和解码器的组合；

对于第K层到第K+1层传递的层间向量为第k层采用的自编码器的隐藏层的表述，其连接矩阵为输入节点的解码器、原始连接的权重矩阵和输出节点的编码器合并而成。

7.根据权利要求4所述的硬件神经网络转换方法，在神经网络应用中存在特殊函数且神经网络硬件芯片不支持该特殊函数的情况下，还包括在完全展开之前：

为所述特殊函数构造专门的神经网络。

8.根据权利要求3所述的硬件神经网络转换方法，所述针对重建的网络拓扑，进行权重参数确定包括：

根据原神经网络的权重初始化通过重建网络拓扑得到的网络的权重；以及

进行权重参数的微调，使得权重满足硬件的权重约束。

9.根据权利要求8所述的硬件神经网络转换方法，所述进行权重参数的微调，使得权重满足硬件的权重约束包括：

(1)首先使用浮点精度表示权重，对构造出的网络进行重新训练，使得与原网络的误差尽可能小；

(2)在神经网络硬件芯片存在可配参数P的情况下，根据第(1)步训练得到的参数，利用EM算法确定一个最好的P和k_{ij}，将所有的权重参数表示为P的函数，重新训练以调节P，其中P为硬件抽象的可配参数，k_{ij}为各个矩阵元在集合S^P中取值的索引；

(3)在神经网络硬件芯片的权重精度低于预定阈值的情况下，固定第(2)步训练得到的P，将所有权重初始化为对应的$S^P_{k_{ij}}$，重新训练以调节k_{ij}，所有权重使用浮点精度存储，但在训练的前馈过程中，所有的权重参数四舍五入到S^P中最接近的值，然后带入前馈计算，而在反馈和更新权重时，仍然使用浮点精度，更新浮点精度的权重值，

其中，将神经网络硬件的基本模块的权重矩阵W取值范围看作一个集合S^P，集合中每个元素都是关于参数P的函数，其中P为硬件可以配置的参数，权重矩阵中的每个元素W_{ij}能够独立地从S^P中选择，即能够独立配置索引k_{ij}，使得$W_{ij} = S^P_{k_{ij}}$，因此权重矩阵W能够配置的是集合参数P和各个权重在集合中取值的索引k_{ij}。

10.根据权利要求1到7任一项所述的硬件神经网络转换方法，所述将每个神经网络基

本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络包括：

在神经网络连接图为有向无环图的情况下，按照神经网络连接图的拓扑序，逐个转换各个神经网络基本单元；

在神经网络连接图为有环有向图的情况下，首先将有环有向图的环拆开，使得神经网络连接图变成有向无环图，然后按照有向无环图的拓扑序，逐个转换各个神经网络基本单元；

按照所述拓扑序，进行转换后的各个神经网络基本单元的训练，其中重新训练所需要的训练数据来源为：训练输入数据是训练样本在经过拓扑序在前的基本单元硬件网络之后产生的输出，训练输出数据是训练样本在原神经网络应用对应层产生的输出。

11.根据权利要求1到7任一项所述的硬件神经网络转换方法，

在神经网络应用为SNN时，在神经网络基本单元转换步骤中使用的训练数据如下获得：对原始网络以稳定频率的电脉冲作为输入，记录各个神经元的电脉冲发放频率，以此作为神经网络基本单元转换步骤中使用的训练数据。

12.根据权利要求1到7任一项所述的硬件神经网络转换方法，在神经网络硬件芯片涉及的神经网络为SNN类型时，根据SNN的神经元模型，推导出SNN在脉冲发放率上的函数关系，基于这个函数关系连续、可导，使用反向传播算法进行训练。

13.一种计算装置，用于将神经网络应用转换为满足硬件约束条件的硬件神经网络，包括存储器和处理器，存储器中存储有计算机可执行指令，当处理器执行所述计算机可执行指令时，执行下述方法：

神经网络连接图获得步骤，获得神经网络应用对应的神经网络连接图，神经网络连接图是一个有向图，图中的每个节点表示一层神经元，每条边表示层间的连接关系；

神经网络连接图拆分步骤，将神经网络连接图拆分为神经网络基本单元，每个神经网络基本单元中，只有入节点和出节点，不存在中间层节点，入节点和出节点之间全相联，而且入节点中的神经元的所有出度在该基本单元内，出节点中的每个神经元的所有入度在该基本单元内；

神经网络基本单元转换步骤，将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络，称之为基本单元硬件网络，一个神经网络基本单元对应于一个或多个神经网络硬件的基本模块虚拟体，每个神经网络硬件的基本模块虚拟体均满足神经网络硬件的基本模块的连接度约束条件，且能够直接映射到神经网络硬件的基本模块；

基本单元硬件网络连接步骤，将得到的基本单元硬件网络按照拆分的顺序连接起来，生成硬件神经网络的参数文件。

14.根据权利要求13所述的计算装置，所执行的方法还包括，在神经网络应用具有卷积层的情况下，在神经网络连接图拆分步骤之前，对于神经网络应用的卷积层进行网络压缩，包括：

获得每一卷积层的多个特征图；

利用DPP提取多样性子集的方法，将这些特征图在所有样本上产生的输出之间的相似性作为DPP算法相关联的矩阵元，利用DPP得到多样性最高的子集，保留该子集，丢弃掉其他特征图节点，将丢弃的特征图所对应的向量投影到保留的特征图所张成的线性空间中，用

丢弃的特征图的投影长度与其原向量长度的比值作为加权系数，将丢弃的特征图与下一层神经元的连接权重加权累加到保留的特征图与下一层神经元的连接权重上。

15.根据权利要求13所述的计算装置，所述神经网络基本单元转换步骤包括：

对每个神经网络基本单元重建网络拓扑；以及

针对重建的网络拓扑，进行权重参数确定。

16.根据权利要求15所述的计算装置，重建网络拓扑包括完全展开操作，经过完全展开，神经网络基本单元被分解为了基本模块虚拟体之间的相互连接，所述完全展开操作包括：

在神经网络基本单元相关联的第一规模的矩阵乘法和/或卷积的大矩阵操作超过了神经网络硬件的基本模块支持的第二规模的小矩阵操作的情况下，执行下述操作：

将第一规模的大矩阵操作拆分为第三数目个第二规模的小矩阵操作，每个小矩阵操作由一个基本模块虚拟体完成；

将针对第一规模的大矩阵操作的输入数据分解为第三数目份，并传送给该第三数目个第二规模的小矩阵操作，此为多播操作；

将来自第三数目个第二规模的小矩阵操作的运算结果汇总为等价于第一规模的大矩阵操作的运算结果，此为归约操作，

在神经网络硬件芯片具有支持多播操作的第一额外模块的情况下，将多播操作分配为由所述第一额外模块虚拟体来执行，否则由多播操作由第一组基本模块虚拟体来完成；

在神经网络硬件芯片具有支持归约操作的第二额外模块的情况下，将归约操作分配为由所述第二额外模块虚拟体来执行，否则由多播操作由第二组基本模块虚拟体来完成。

17.根据权利要求16所述的计算装置，在神经网络硬件芯片上基本模块数额不足的情况下，采用时分方法来利用基本模块。

18.根据权利要求16所述的计算装置，重建网络拓扑还包括在完全展开操作之前进行重编码操作，包括：

利用自编码器进行层间数据重编码，自编码器是神经网络，由3层神经元组成，包括输入层、隐藏层和输出层，其中输出层的节点数和输入层节点数相同，隐藏层的节点数量多于层间向量数据的维度，训练该网络，使得输出层的值与输入层的值尽可能相近，其中输入层和输出层的精度为神经网络应用的精度，隐藏层采用神经网络硬件基本模块之间的传输数据的精度，自编码器被转换为编码器和解码器的组合；

对于第K层到第K+1层传递的层间向量为第k层采用的自编码器的隐藏层的表述，其连接矩阵为输入节点的解码器、原始连接的权重矩阵和输出节点的编码器合并而成。

19.根据权利要求16所述的计算装置，在神经网络应用中存在特殊函数且神经网络硬件芯片不支持该特殊函数的情况下，还包括在完全展开之前：

为所述特殊函数构造专门的神经网络。

20.根据权利要求15所述的计算装置，所述针对重建的网络拓扑，进行权重参数确定包括：

根据原神经网络的权重初始化通过重建网络拓扑得到的网络的权重；以及

进行权重参数的微调，使得权重满足硬件的权重约束。

21.根据权利要求20所述的计算装置，所述进行权重参数的微调，使得权重满足硬件的

权重约束包括：

(1)首先使用浮点精度表示权重，对构造出的网络进行重新训练，使得与原网络的误差尽可能小；

(2)在神经网络硬件芯片存在可配参数P的情况下，根据第(1)步训练得到的参数，利用EM算法确定一个最好的P和k_{ij}，将所有的权重参数表示为P的函数，重新训练以调节P，其中P为硬件抽象的可配参数，k_{ij}为各个矩阵元在集合S^P中取值的索引；

(3)在神经网络硬件芯片的权重精度低于预定阈值的情况下，固定第(2)步训练得到的P，将所有权重初始化为对应的$S^P_{k_{ij}}$，重新训练以调节k_{ij}，所有权重使用浮点精度存储，但在训练的前馈过程中，所有的权重参数四舍五入到S^P中最接近的值，然后带入前馈计算，而在反馈和更新权重时，仍然使用浮点精度，更新浮点精度的权重值，

其中，将神经网络硬件的基本模块的权重矩阵W取值范围看作一个集合S^P，集合中每个元素都是关于参数P的函数，其中P为硬件可以配置的参数，权重矩阵中的每个元素W_{ij}能够独立地从S^P中选择，即能够独立配置索引k_{ij}，使得$W_{ij} = S^P_{k_{ij}}$，因此权重矩阵W能够配置的是集合参数P和各个权重在集合中取值的索引k_{ij}。

22.根据权利要求13到19任一项所述的计算装置，所述将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络包括：

在神经网络连接图为有向无环图的情况下，按照神经网络连接图的拓扑序，逐个转换各个神经网络基本单元；

在神经网络连接图为有环有向图的情况下，首先将有环有向图的环拆开，使得神经网络连接图变成有向无环图，然后按照有向无环图的拓扑序，逐个转换各个神经网络基本单元；

按照所述拓扑序，进行转换后的各个神经网络基本单元的训练，其中重新训练所需要的训练数据来源为：训练输入数据是训练样本在经过拓扑序在前的基本单元硬件网络之后产生的输出，训练输出数据是训练样本在原神经网络应用对应层产生的输出。

23.根据权利要求13到19任一项所述的计算装置，

在神经网络应用为SNN时，在神经网络基本单元转换步骤中使用的训练数据如下获得：对原始网络以稳定频率的电脉冲作为输入，记录各个神经元的电脉冲发放频率，以此作为神经网络基本单元转换步骤中使用的训练数据。

24.根据权利要求13到19任一项所述的计算装置，在神经网络硬件芯片涉及的神经网络为SNN类型时，根据SNN的神经元模型，推导出SNN在脉冲发放率上的函数关系，基于这个函数关系连续、可导，使用反向传播算法进行训练。

25.一种将神经网络软件应用编译为硬件神经网络的编译方法，包括：

获得神经网络软件应用和神经网络硬件芯片的配置情况；

基于神经网络硬件的配置情况，将神经网络软件应用转换硬件神经网络，所述硬件神经网络由神经网络硬件芯片的基本模块连接而成；

输出硬件神经网络的参数文件，所述参数文件描述所述基本模块之间的连接关系以及各个基本模块的参数配置情况。

26.一种神经网络软硬件协作系统，包括：

神经网络硬件芯片，神经网络硬件芯片上具有基本模块，基本模块以硬件形式执行矩阵向量乘和激活函数的操作，神经网络硬件芯片上的基本模块的参数和基本模块之间的连接能够由确定格式的配置文件配置；

编译层单元，用于将神经网络应用编译为硬件神经网络的参数文件，基于参数文件能够将硬件神经网络映射到一个或多个神经网络硬件芯片，映射后的一个或多个神经网络硬件芯片能够运行所述神经网络应用的功能。

27.根据权利要求26所述的神经网络软硬件协作系统，

所述编译层单元配置为执行下述方法：

硬件配置数据获得步骤，获得神经网络硬件芯片的配置情况数据；

神经网络连接图获得步骤，获得神经网络应用对应的神经网络连接图，神经网络连接图是一个有向图，图中的每个节点表示一层神经元，每条边表示层间的连接关系；

神经网络连接图拆分步骤，将神经网络连接图拆分为神经网络基本单元，每个神经网络基本单元中，只有入节点和出节点，不存在中间层节点，入节点和出节点之间全相联，而且入节点中的神经元的所有出度在该基本单元内，出节点中的每个神经元的所有入度在该基本单元内；

神经网络基本单元转换步骤，将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络，称之为基本单元硬件网络，一个神经网络基本单元对应于一个或多个神经网络硬件的基本模块虚拟体，每个神经网络硬件的基本模块虚拟体均满足神经网络硬件的基本模块的连接度约束条件，且能够直接映射到神经网络硬件的基本模块；

基本单元硬件网络连接步骤，将得到的基本单元硬件网络按照拆分的顺序连接起来，生成硬件神经网络的参数文件。

硬件神经网络转换方法、计算装置、软硬件协作系统

技术领域

[0001] 本发明总体地涉及神经网络技术领域，更具体地涉及由神经网络芯片来实现软件神经网络的技术。

背景技术

[0002] 最近几年，深度学习技术取得了突破性进展，在图像识别、语言识别、自然语言处理等诸多领域均取得了很高的准确率，但深度学习需要海量计算资源，传统的通用处理器已经很难满足深度学习的计算需求，将深度学习硬件化，为其设计专用芯片已经成为了一个重要的发展方向。与此同时，随着脑科学的发展，由于大脑相比传统的冯诺依曼计算机，具有超低功耗，高容错性等特点，且在处理非结构化信息和智能任务方面具有显著优势，借鉴大脑的计算模式构建新型的类脑计算系统和类脑计算芯片已经成为一个新兴的发展方向。

[0003] 无论是深度学习还是类脑计算，其底层的计算模型均是神经网络(NeuralNetwork，NN)，主要区别在于，深度学习使用的主要是人工神经网络(ArtificialNeuralNetwork，ANN)，而类脑计算主要使用的是脉冲神经网络(SpikingNeuralNetwork，SNN)，两者基本组成单元均为神经元，由大量神经元相互连接成网络。神经元之间的连接可以看作带权重的有向边，神经元的输出会被神经元之间的连接所加权，然后传递给所连到的神经元，而每个神经元接收到的所有输入会被累加起来进行进一步处理，产生神经元的输出。ANN和SNN的主要区别在于，ANN的神经元输出的是数值，与边权相乘进行加权；而SNN的神经元输出的是一个个电脉冲信号，电脉冲信号经过加权成为不同强度的电流信号；ANN的神经元对于其他神经元的输入，会经过一个激活函数直接算出神经元的输出值；而SNN的神经元接收到其他神经元输入的电流信号，会根据其神经元模型更新其状态，当达到特定状态便会发放一个电脉冲，并重置状态。

[0004] 神经网络的建模通常以若干神经元为一层，层与层之间相互连接来构建，图10所示的是一种链状的神经网络，图中每一个圆表示一个神经元，每一个箭头表示神经元之间的连接，每个连接均有权重，实际神经网络的结构不限于链状的网络结构。

[0005] 神经网络的核心计算是矩阵向量乘操作，包含n个神经元的层L_n产生的输出可以用长度为n的向量V_n表示，与包含m个神经元的层L_m全相联，连接权重可以表示成矩阵$M_{n\times m}$，矩阵大小为n行m列，每个矩阵元素表示一个连接的权重。则加权之后输入到L_m的向量为$M_{n\times m}V_n$，这样的矩阵向量乘法运算是神经网络最核心的计算。

[0006] 由于矩阵向量乘计算量非常大，在传统的通用处理器上进行大量的矩阵乘运算需要耗费大量的时间，因此神经网络加速芯片和类脑芯片也都是以加速矩阵乘法运算为主要的设计目标，在具体实现上，通常是用硬件实现一定规模的矩阵向量乘法模块(例如实现大小为256×256的矩阵与长度为256的向量相乘的基本模块)，然后用片上网络(NetworkonChip，NoC)等技术将基本模块连接起来。通过将矩阵向量乘法硬件化，运算速度可以大大提高。

[0007] 然而硬件化也约束了其所能支持的神经网络应用的自由度，这也带来一个重要的问题：很难使用这样的芯片来运行实际的神经网络应用。虽然神经网络芯片可以高效地进行矩阵向量乘法运算，但神经网络应用与底层芯片之间仍然存在很大的不同，例如：

[0008] (1)神经网络硬件的基本模块通常是固定规模的矩阵向量乘，而实际神经网络应用中矩阵运算的规模是任意的。

[0009] (2)神经网络应用通常使用32位浮点数进行计算，而硬件有时会设计成较低的精度，甚至整数来进行计算以提高效率。

[0010] (3)神经网络硬件的激活函数(对ANN而言)或神经元模型(对于SNN而言)通常是固定的，而神经网络应用的激活函数或神经元模型通常非常灵活，且不断会有新的激活函数和神经元模型被引入到神经网络应用中。

[0011] 下面概况一下现有技术的硬件芯片系列。

[0012] 1、现有技术之一：寒武纪芯片系列

[0013] 1(1)现有技术一的技术方案

[0014] 寒武纪芯片的计算核心通过高速的三级流水线实现了16×16规模的矩阵向量乘法运算和非线性激活函数，芯片上还配置了3块专用的存储模块，分别用于存放输入数据、输出数据和权重数据，通过控制器从片上存储模块中调取数据送入计算核心进行计算。对于更大规模的矩阵运算，例如32×32的矩阵，该技术方案会将其拆分成4个16×16的矩阵，通过控制器依次载入到计算核心中完成计算，最后再将计算结果累加合并起来。通过对计算核心的时分复用，完成对任意规模神经网络的支持。另一方面，在寒武纪芯片计算核心的第三级流水步中，提供了多种常见激活函数，以支持绝大多数神经网络应用。

[0015] 1(2)现有技术一的缺点

[0016] 寒武纪芯片的做法将神经网络的权重同计算核心分离开，通过软件来控制计算资源的时分复用和存储器的访问，由于该方法还是将计算和存储分离，本质上还是冯诺依曼架构下的一种定制方案，仍然需要在计算单元和存储单元之间来回传输权重数据，仍然会受制于冯诺依曼瓶颈。虽然寒武纪芯片在提高计算核心与存储单元之间的带宽上做了很大的努力，但随着神经网络应用规模的增加，权重数据的访问终将成为系统瓶颈。

[0017] 且由于计算逻辑和片上存储开销较大，芯片集成度无法做到很高，每块芯片上集成的计算核心数量非常有限。

[0018] 2、与本发明相关的现有技术二：TrueNorth芯片

[0019] 2(1)、现有技术二的技术方案

[0020] TrueNorth是IBM公司的神经形态芯片，每块芯片上集成了4096个神经突触核，每个神经突触核可以处理256×256的神经突触计算(即矩阵向量乘法运算)。为了提高集成度，TrueNorth的神经突触核进行了极大的精简，采用了非常简单的LeakyIntegrateandFire(LIF)神经元模型(一种常用的SNN神经元模型)，对权重也进行了极大的压缩，每个神经元至多只能拥有256个输入突触，且这256个输入突触的权重也只有3个可选的值。

[0021] 为了运用TrueNorth运行实际的神经网络，IBM设计了一套Corelet语言来对TrueNorth进行编程，将大的任务逐步分解成小的任务之间的连接，使得最小的任务刚好能在神经突触核上。Corelet将硬件的种种约束暴露给应用层，在设计神经网络的时候需要考

虑TrueNorth硬件本身的约束。

[0022] 2(2)现有技术二的缺点

[0023] 在TrueNorth的芯片设计中，为了提高芯片的集成度，在有限的面积内放置更多的神经突触核，TrueNorth芯片的神经突触核对神经网络有很强的约束。因此很难将现有的神经网络应用放到TrueNorth芯片上运行，对于各种智能任务，需要重新设计、训练一个专门针对TrueNorth芯片的神经网络，且由于硬件对应用层约束，针对TrueNorth重新设计、训练的神经网络目前很难在图像识别等领域达到与目前最先进的神经网络相当的准确率。

[0024] 3、与本发明相关的现有技术三:新型器件——忆阻器

[0025] 3(1)、现有技术三的技术方案

[0026] 忆阻器是一种新型的半导体器件，其电阻阻值可以在特定的输入电流下改变。忆阻器的阻值可以用来存储数据，相比传统的DRAM(动态随机存储器)和SRAM(静态随机存储器)具有存储密度高的特点，且由于其数据是通过阻值来存储的，在失去供电的情况下也不会丢失数据。此外，忆阻器也可以进行计算，是一种计算与存储融合的理想器件。

[0027] 图11示出了基于忆阻器的交叉开关(Crossbar)结构的示意图。

[0028] 如图11所示，通过将线路排布成交叉开关(Crossbar)，并在相交点用忆阻器相连，将忆阻器的电导值(电阻的倒数)设置为权重矩阵的矩阵元数值，通过在输入端输入电压值，在输出端即可完成矩阵向量乘法运算。以此为基本单元，可以构建基于新型器件的神经形态芯片。由于其集成度很高，计算和存储融合的特点，无需来回传输权重数据，在构建大规模神经形态芯片上具有很大的潜力。

[0029] 3(2)技术方案三的缺点

[0030] 由于忆阻器的计算基于模拟电路，其模拟信号可以达到的精度是有限的，权重的取值范围也取决于忆阻器的阻变范围。且同TrueNorth一样有连接度的约束限制，很难直接将现有的神经网络直接放到上面运行。

[0031] 总结起来，现有技术方案一寒武纪芯片致力于让芯片适配神经网络应用的需求，通过时分复用的方式，使得芯片可以支持任意规模的神经网络，通过内置常用激活函数来支持现有的神经网络。一方面，由于其存储和计算分离的特点，始终受制于冯诺依曼瓶颈，随着应用规模的扩大，其效率将受制于存储与计算之间的传输带宽;另一方面，由于其固化了常用的激活函数，随着神经网络应用技术的发展，新的激活函数和神经元模型需要芯片不断适应应用的发展而进行相应的修改;并且，由于其芯片自由度较高，逻辑相对复杂，无法做到很高的集成度。现有技术方案二TrueNorth致力于将应用适配神经网络芯片，而底层芯片则致力于提高集成度和效率，降低功耗。通过简化其所支持的神经元模型，使得在很小的芯片面积和极低的功耗下继承了数百万神经元。且可以同技术方案三结合，使用新型器件和工艺进一步提高集成度。但这一类方案在应用上提出了太多的约束，无法与现有应用很好的结合，也很难在复杂任务上取得与目前最先进的神经网络相当的效果。

[0032] 可见，现有的神经网络硬件通常直接与神经网络应用衔接，要么会出现硬件过于简单，约束了应用的自由度的问题，要么会出现硬件自由度高，比较复杂，从而很难提高集成度和效率的问题。

[0033] 需要更普适性的将任意神经网络应用适配到任意神经网络芯片上的通用技术。

发明内容

[0034] 鉴于上述情况，做出了本发明。

[0035] 根据本发明的一个方面，提供了一种将神经网络应用转换为满足硬件约束条件的硬件神经网络的硬件神经网络转换方法，可以包括：神经网络连接图获得步骤，获得神经网络应用对应的神经网络连接图，神经网络连接图是一个有向图，图中的每个节点表示一层神经元，每条边表示层间的连接关系；神经网络连接图拆分步骤，将神经网络连接图拆分为神经网络基本单元，每个神经网络基本单元中，只有入节点和出节点，不存在中间层节点，入节点和出节点之间全相联，，而且入节点中的神经元的所有出度在该基本单元内，出节点中的每个神经元的所有入度在该基本单元内；神经网络基本单元转换步骤，将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络，称之为基本单元硬件网络，一个神经网络基本单元对应于一个或多个神经网络硬件的基本模块虚拟体，每个神经网络硬件的基本模块虚拟体均满足神经网络硬件的基本模块的连接度约束条件，且能够直接映射到神经网络硬件的基本模块；基本单元硬件网络连接步骤，将得到的基本单元硬件网络按照拆分的顺序连接起来，生成硬件神经网络的参数文件。

[0036] 根据上述硬件神经网络转换方法，还可以包括，在神经网络应用具有卷积层的情况下，在神经网络连接图拆分步骤之前，对于神经网络应用的卷积层进行网络压缩，网络压缩操作可以包括：获得每一卷积层的多个特征图；利用DPP提取多样性子集的方法，将这些特征图在所有样本上产生的输出之间的相似性作为DPP算法相关联的矩阵元，利用DPP得到多样性最高的子集，保留该子集，丢弃掉其他特征图节点，将丢弃的特征图所对应的向量投影到保留的特征图所张成的线性空间中，用丢弃的特征图的投影长度与其原向量长度的比值作为加权系数，将丢弃的特征图与下一层神经元的连接权重加权累加到保留的特征图与下一层神经元的连接权重上。

[0037] 根据上述硬件神经网络转换方法，所述神经网络基本单元转换步骤包括：对每个神经网络基本单元重建网络拓扑；以及针对重建的网络拓扑，进行权重参数确定。

[0038] 根据上述硬件神经网络转换方法，重建网络拓扑包括完全展开操作，经过完全展开，神经网络基本单元被分解为了基本模块虚拟体之间的相互连接，所述完全展开操作包括：在神经网络基本单元相关联的第一规模的矩阵乘法和/或卷积的大矩阵操作超过了神经网络硬件的基本模块支持的第二规模的小矩阵操作的情况下，执行下述操作：将第一规模的大矩阵操作拆分为第三数目个第二规模的小矩阵操作，每个小矩阵操作由一个基本模块虚拟体完成；将针对第一规模的大矩阵操作的输入数据分解为第三数目份，并传送给该第三数目个第二规模的小矩阵操作，此为多播操作；将来自第三数目个第二规模的小矩阵操作的运算结果汇总为等价于第一规模的大矩阵操作的运算结果，此为归约操作，在神经网络硬件芯片具有支持多播操作的第一额外模块的情况下，将多播操作分配为由所述第一额外模块虚拟体来执行，否则由多播操作由第一组基本模块虚拟体来完成；在神经网络硬件芯片具有支持归约操作的第二额外模块的情况下，将归约操作分配为由所述第二额外模块虚拟体来执行，否则由多播操作由第二组基本模块虚拟体来完成。

[0039] 根据上述硬件神经网络转换方法，在神经网络硬件芯片上基本模块数额不足的情况下，采用时分方法来利用基本模块。

[0040] 根据上述硬件神经网络转换方法，重建网络拓扑还包括在完全展开操作之前进行

重编码操作，可以包括：利用自编码器进行层间数据重编码，自编码器是神经网络，由3层神经元组成，包括输入层、隐藏层和输出层，其中输出层的节点数和输入层节点数相同，隐藏层的节点数量多于层间向量数据的维度，训练该网络，使得输出层的值与输入层的值尽可能相近，其中输入层和输出层的精度为神经网络应用的精度，隐藏层采用神经网络硬件基本模块之间的传输数据的精度，自编码器被转换为编码器和解码器的组合；对于第K层到第K+1层传递的层间向量为第k层采用的自编码器的隐藏层的表述，其连接矩阵为输入节点的解码器、原始连接的权重矩阵和输出节点的编码器合并而成。

[0041] 根据上述硬件神经网络转换方法，在神经网络应用中存在特殊函数且神经网络硬件芯片不支持该特殊函数的情况下，还包括在完全展开之前为所述特殊函数构造专门的神经网络。

[0042] 根据上述硬件神经网络转换方法，所述针对重建的网络拓扑，进行权重参数确定包括：根据原神经网络的权重初始化通过重建网络拓扑得到的网络的权重；以及进行权重参数的微调，使得权重满足硬件的权重约束。根据上述硬件神经网络转换方法，所述进行权重参数的微调，使得权重满足硬件的权重约束包括：(1)首先使用浮点精度表示权重，对构造出的网络进行重新训练，使得与原网络的误差尽可能小；(2)在神经网络硬件芯片存在可配参数P的情况下，根据第(1)步训练得到的参数，利用EM算法确定一个最好的P和k_{ij}，将所有的权重参数表示为P的函数，重新训练以调节P，其中P为硬件抽象的可配参数，k_{ij}为各个矩阵元在集合S^P中取值的索引；(3)在神经网络硬件芯片的权重精度低于预定阈值的情况下，固定第(2)步训练得到的P，将所有权重初始化为对应的$S^P_{k_{ij}}$，重新训练以调节k_{ij}，所有权重使用浮点精度存储，但在训练的前馈过程中，所有的权重参数四舍五入到S^P中最接近的值，然后带入前馈计算，而在反馈和更新权重时，仍然使用浮点精度，更新浮点精度的权重值，其中，将神经网络硬件的基本模块的权重矩阵W取值范围看作一个集合S^P，集合中每个元素都是关于参数P的函数，其中P为硬件可以配置的参数，权重矩阵中的每个元素W_{ij}能够独立地从S^P中选择，即能够独立配置索引k_{ij}，使得$W_{ij} = S^P_{k_{ij}}$，因此权重矩阵W能够配置的是集合参数P和各个权重在集合中取值的索引k_{ij}。

[0043] 根据上述硬件神经网络转换方法，所述将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络可以包括：在神经网络连接图为有向无环图的情况下，按照神经网络连接图的拓扑序，逐个转换各个神经网络基本单元；在神经网络连接图为有环有向图的情况下，首先将有环有向图的环拆开，使得神经网络连接图变成有向无环图，然后按照有向无环图的拓扑序，逐个转换各个神经网络基本单元；按照所述拓扑序，进行转换后的各个神经网络基本单元的训练，其中重新训练所需要的训练数据来源为：训练输入数据是训练样本在经过拓扑序在前的基本单元硬件网络之后产生的输出，训练输出数据是训练样本在原神经网络应用对应层产生的输出。

[0044] 根据上述硬件神经网络转换方法，在神经网络应用为SNN时，在神经网络基本单元转换步骤中使用的训练数据如下获得：对原始网络以稳定频率的电脉冲作为输入，记录各个神经元的电脉冲发放频率，以此作为神经网络基本单元转换步骤中使用的训练数据。

[0045] 根据上述硬件神经网络转换方法，在神经网络硬件芯片涉及的神经网络为SNN类型时，根据SNN的神经元模型，推导出SNN在脉冲发放率上的函数关系，基于这个函数关系连

续、可导，使用反向传播算法进行训练。

[0046] 根据本发明的另一方面，提供了一种计算装置，用于将神经网络应用转换为满足硬件约束条件的硬件神经网络，包括存储器和处理器，存储器中存储有计算机可执行指令，当处理器执行所述计算机可执行指令时，执行下述方法：神经网络连接图获得步骤，获得神经网络应用对应的神经网络连接图，神经网络连接图是一个有向图，图中的每个节点表示一层神经元，每条边表示层间的连接关系；神经网络连接图拆分步骤，将神经网络连接图拆分为神经网络基本单元，每个神经网络基本单元中，只有入节点和出节点，不存在中间层节点，入节点和出节点之间全相联，而且入节点中的神经元的所有出度在该基本单元内，出节点中的每个神经元的所有入度在该基本单元内；神经网络基本单元转换步骤，将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络，称之为基本单元硬件网络，一个神经网络基本单元对应于一个或多个神经网络硬件的基本模块虚拟体，每个神经网络硬件的基本模块虚拟体均满足神经网络硬件的基本模块的连接度约束条件，且能够直接映射到神经网络硬件的基本模块；基本单元硬件网络连接步骤，将得到的基本单元硬件网络按照拆分的顺序连接起来，生成硬件神经网络的参数文件。

[0047] 根据上述计算装置，所执行的方法还包括，在神经网络应用具有卷积层的情况下，在神经网络连接图拆分步骤之前，对于神经网络应用的卷积层进行网络压缩，包括：获得每一卷积层的多个特征图；利用DPP提取多样性子集的方法，将这些特征图在所有样本上产生的输出之间的相似性作为DPP算法相关联的矩阵元，利用DPP得到多样性最高的子集，保留该子集，丢弃掉其他特征图节点，将丢弃的特征图所对应的向量投影到保留的特征图所张成的线性空间中，用丢弃的特征图的投影长度与其原向量长度的比值作为加权系数，将丢弃的特征图与下一层神经元的连接权重加权累加到保留的特征图与下一层神经元的连接权重上。

[0048] 根据上述计算装置，所述神经网络基本单元转换步骤可以包括：对每个神经网络基本单元重建网络拓扑；以及针对重建的网络拓扑，进行权重参数确定。

[0049] 根据上述计算装置，重建网络拓扑包括完全展开操作，经过完全展开，神经网络基本单元被分解为了基本模块虚拟体之间的相互连接，所述完全展开操作包括：

[0050] 在神经网络基本单元相关联的第一规模的矩阵乘法和/或卷积的大矩阵操作超过了神经网络硬件的基本模块支持的第二规模的小矩阵操作的情况下，执行下述操作：将第一规模的大矩阵操作拆分为第三数目个第二规模的小矩阵操作，每个小矩阵操作由一个基本模块虚拟体完成；将针对第一规模的大矩阵操作的输入数据分解为第三数目份，并传送给该第三数目个第二规模的小矩阵操作，此为多播操作；将来自第三数目个第二规模的小矩阵操作的运算结果汇总为等价于第一规模的大矩阵操作的运算结果，此为归约操作，在神经网络硬件芯片具有支持多播操作的第一额外模块的情况下，将多播操作分配为由所述第一额外模块虚拟体来执行，否则由多播操作由第一组基本模块虚拟体来完成；在神经网络硬件芯片具有支持归约操作的第二额外模块的情况下，将归约操作分配为由所述第二额外模块虚拟体来执行，否则由多播操作由第二组基本模块虚拟体来完成。

[0051] 根据上述计算装置，在神经网络硬件芯片上基本模块数额不足的情况下，采用时分方法来利用基本模块。

[0052] 根据上述计算装置，重建网络拓扑还包括在完全展开操作之前进行重编码操作，

包括:利用自编码器进行层间数据重编码,自编码器是神经网络,由3层神经元组成,包括输入层、隐藏层和输出层,其中输出层的节点数和输入层节点数相同,隐藏层的节点数量多于层间向量数据的维度,训练该网络,使得输出层的值与输入层的值尽可能相近,其中输入层和输出层的精度为神经网络应用的精度,隐藏层采用神经网络硬件基本模块之间的传输数据的精度,自编码器被转换为编码器和解码器的组合;对于第K层到第K+1层传递的层间向量为第k层采用的自编码器的隐藏层的表述,其连接矩阵为输入节点的解码器、原始连接的权重矩阵和输出节点的编码器合并而成。

[0053] 根据上述计算装置,在神经网络应用中存在特殊函数且神经网络硬件芯片不支持该特殊函数的情况下,还包括在完全展开之前:为所述特殊函数构造专门的神经网络。

[0054] 根据上述计算装置,所述针对重建的网络拓扑,进行权重参数确定包括:根据原神经网络的权重初始化通过重建网络拓扑得到的网络的权重;以及进行权重参数的微调,使得权重满足硬件的权重约束。

[0055] 根据上述计算装置,所述进行权重参数的微调,使得权重满足硬件的权重约束包括:(1)首先使用浮点精度表示权重,对构造出的网络进行重新训练,使得与原网络的误差尽可能小;(2)在神经网络硬件芯片存在可配参数P的情况下,根据第(1)步训练得到的参数,利用EM算法确定一个最好的P和k_{ij},将所有的权重参数表示为P的函数,重新训练以调节P,其中P为硬件抽象的可配参数,k_{ij}为各个矩阵元在集合S^P中取值的索引;(3)在神经网络硬件芯片的权重精度低于预定阈值的情况下,固定第(2)步训练得到的P,将所有权重初始化为对应的$S^P_{k_{ij}}$,重新训练以调节k_{ij},所有权重使用浮点精度存储,但在训练的前馈过程中,所有的权重参数四舍五入到S^P中最接近的值,然后带入前馈计算,而在反馈和更新权重时,仍然使用浮点精度,更新浮点精度的权重值,其中,将神经网络硬件的基本模块的权重矩阵W取值范围看作一个集合S^P,集合中每个元素都是关于参数P的函数,其中P为硬件可以配置的参数,权重矩阵中的每个元素W_{ij}能够独立地从S^P中选择,即能够独立配置索引k_{ij},使得$W_{ij} = S^P_{k_{ij}}$,因此权重矩阵W能够配置的是集合参数P和各个权重在集合中取值的索引k_{ij}。

[0056] 根据上述计算装置,所述将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络包括:在神经网络连接图为有向无环图的情况下,按照神经网络连接图的拓扑序,逐个转换各个神经网络基本单元;在神经网络连接图为有环有向图的情况下,首先将有环有向图的环拆开,使得神经网络连接图变成有向无环图,然后按照有向无环图的拓扑序,逐个转换各个神经网络基本单元;按照所述拓扑序,进行转换后的各个神经网络基本单元的训练,其中重新训练所需要的训练数据来源为:训练输入数据是训练样本在经过拓扑序在前的基本单元硬件网络之后产生的输出,训练输出数据是训练样本在原神经网络应用对应层产生的输出。

[0057] 根据上述计算装置,在神经网络应用为SNN时,在神经网络基本单元转换步骤中使用的训练数据如下获得:对原始网络以稳定频率的电脉冲作为输入,记录各个神经元的电脉冲发放频率,以此作为神经网络基本单元转换步骤中使用的训练数据。

[0058] 根据上述计算装置,在神经网络硬件芯片涉及的神经网络为SNN类型时,根据SNN的神经元模型,推导出SNN在脉冲发放率上的函数关系,基于这个函数关系连续、可导,使用

反向传播算法进行训练。

[0059] 根据本发明的另一方面，提供了一种将神经网络软件应用编译为硬件神经网络的编译方法，可以包括：获得神经网络软件应用和神经网络硬件芯片的配置情况；基于神经网络硬件的配置情况，将神经网络软件应用转换硬件神经网络，所述硬件神经网络由神经网络硬件芯片的基本模块连接而成；输出硬件神经网络的参数文件，所述参数文件描述所述基本模块之间的连接关系以及各个基本模块的参数配置情况。

[0060] 根据本发明的另一方面，提供了一种神经网络软硬件协作系统，可以包括：神经网络硬件芯片，神经网络硬件芯片上具有基本模块，基本模块以硬件形式执行矩阵向量乘和激活函数的操作，神经网络硬件芯片上的基本模块的参数和基本模块之间的连接能够由确定格式的配置文件配置；编译层单元，用于将神经网络应用编译为硬件神经网络的参数文件，基于参数文件能够将硬件神经网络映射到一个或多个神经网络硬件芯片，映射后的一个或多个神经网络硬件芯片能够运行所述神经网络应用的功能。

[0061] 根据上述神经网络软硬件协作系统，所述编译层单元配置为执行下述方法：硬件配置数据获得步骤，获得神经网络硬件芯片的配置情况数据；神经网络连接图获得步骤，获得神经网络应用对应的神经网络连接图，神经网络连接图是一个有向图，图中的每个节点表示一层神经元，每条边表示层间的连接关系；神经网络连接图拆分步骤，将神经网络连接图拆分为神经网络基本单元，每个神经网络基本单元中，只有入节点和出节点，不存在中间层节点，入节点和出节点之间全相联，，而且入节点中的神经元的所有出度在该基本单元内，出节点中的每个神经元的所有入度在该基本单元内；神经网络基本单元转换步骤，将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络，称之为基本单元硬件网络，一个神经网络基本单元对应于一个或多个神经网络硬件的基本模块虚拟体，每个神经网络硬件的基本模块虚拟体均满足神经网络硬件的基本模块的连接度约束条件，且能够直接映射到神经网络硬件的基本模块；基本单元硬件网络连接步骤，将得到的基本单元硬件网络按照拆分的顺序连接起来，生成硬件神经网络的参数文件。

[0062] 本公开提出了一种全新的神经网络和类脑计算的软硬件体系。

[0063] 如前所述，现有的技术路线均是让神经网络的应用和芯片直接适配，要么将芯片直接去适配应用的自由度，这会带来性能瓶颈；要么将芯片的约束暴露给应用，这约束了应用的能力。相对比，本发明实施例的硬件神经网络转换方法，在神经网络应用和神经网络芯片之间加上了一个中间层，通过一种相当于传统计算机体系当中的编译的技术解决了神经网络应用与神经网络应用芯片之间的适配问题，同时解耦合了应用和芯片的开发。

[0064] 另外，本发明实施例的硬件神经网络转换方法，对于任意的复杂神经网络和满足硬件抽象的任意硬件，提供了一种通用的流程，可以将复杂神经网络转换成满足该硬件约束条件的特定网络，且功能上与原网络基本等效。该流程的核心在于将复杂网络进行分解，由于每个基本单元所做的运算相对简单，转换过程相比直接转换整个网络更有保障能收敛，且收敛速度也更快。

[0065] 而且，本发明实施例的硬件神经网络转换方法，通过对神经网络连接图中的节点进行分组，将神经网络拆分成若干基本单元，使得基本单元内任意一个节点的入边或出边全部在该基本单元内，从而在基本单元内解决了连接度的问题之后，将转换完的基本单元

重新链接起来，得到的网络仍然能满足连接度的要求。

[0066] 另外，在前面所述的一个示例中，按照拓扑序逐个模块进行转换，将前面产生的误差引入到后面的微调中，使得各个基本模块转换引入的误差不会逐层积累。

[0067] 另外，在一个示例中，在神经网络应用具有卷积层的情况下，在神经网络连接图拆分步骤之前，可以对于神经网络应用的卷积层进行网络压缩，缩小网络规模，节省硬件资源。

附图说明

[0068] 从下面结合附图对本发明实施例的详细描述中，本发明的这些和/或其它方面和优点将变得更加清楚并更容易理解，其中：

[0069] 图1示出了根据本发明实施例的硬件神经网络转换技术的应用情境1000的示意图。

[0070] 图2示出了根据本发明实施例的编译层1200执行的硬件神经网络转换方法200的总体流程图。

[0071] 图3给出了神经网络连接图的示例，其中节点1、2、3、4、5的每个表示为一层神经元。

[0072] 图4给出了神经网络基本单元400的示例性示意图。

[0073] 图5(a)-(c)示出了将神经网络连接图拆分为多个神经网络基本单元的过程示意图。

[0074] 图6示出了神经网络基本单元转换中的重建网络拓扑操作和权重参数微调操作的示意图。

[0075] 图7示出了对于三层神经网络利用自编码器重新编码以得到扩充后的三层神经网络的过程示意图。

[0076] 图8示出了max操作的神经网络替换。

[0077] 图9示出了根据本发明一个实施例的对于大规模矩阵乘法操作的完全展开2313的示例性示意图。

[0078] 图10示出了链状的神经网络的示意图。

[0079] 图11示出了基于忆阻器的交叉开关(Crossbar)结构的示意图。

具体实施方式

[0080] 为了使本领域技术人员更好地理解本发明，下面结合附图和具体实施方式对本发明作进一步详细说明。

[0081] 在详细描述各个实施例之前，给出本文中使用的术语的解释。

[0082] 硬件神经网络指满足硬件的约束条件的神经网络。

[0083] 神经网络硬件芯片指以神经网络为目标应用的芯片。

[0084] 神经网络连接图：神经网络连接图是一个有向图，图中的每个节点表示一层神经元，每条边表示层间的连接关系，在ANN神经网络应用情况下，对应的神经网络连接图是无环有向图，在SNN神经网络应用情况下，对应的神经网络连接图是有向有环图。

[0085] 神经网络基本单元：每个神经网络基本单元中，只有入节点和出节点，不存在中间

层节点，入节点和出节点之间全相联，，而且入节点中的神经元的所有出度在该基本单元内，出节点中的每个神经元的所有入度在该基本单元内。

[0086] 神经网络硬件芯片：由大量物理核通过互连系统连接起来，可能有各种各样的拓扑，可以接受一定的配置。

[0087] 物理核：矩阵向量乘+激活函数构成的神经网络硬件基本模块，其功能为接收输入、先做矩阵加权、然后经过激活函数产生输出。

[0088] 硬件神经网络的参数文件：包括描述虚拟核的参数和虚拟核之间的连接关系的信息，虚拟核的参数包括例如连接矩阵等。

[0089] 虚拟核：虚拟核和物理核对应，是物理核的一个抽象，在本文中是算法最后得到的一个连接图当中的一个个硬件基本模块虚拟体。转换算法结束之后得到了一堆虚拟核以及相互之间的连接关系，然后要通过映射算法把虚拟核布到神经网络硬件芯片的物理核上。

[0090] 映射：将虚拟核布局到物理核上的过程。

[0091] 连接度约束：每个神经网络硬件基本模块只能支持固定规模的矩阵运算，所以神经元的入度不能超过硬件基本模块的输入数量，神经元的出度不能超过硬件基本模块的出度。另外还有一点就是硬件基本模块之间的连接只支持一对一的连接，即硬件基本模块的一个输出只能发送给另一个硬件基本模块的输入，这也是连接度的约束，不过不是所有神经网络硬件都有这个约束。

[0092] 本公开提出了在硬件和应用之间引入中间层的思路，并提出了一种将任意神经网络(无论是ANN还是SNN)透明地转换并适配到任意神经网络芯片上的通用方法和流程，类似于传统计算机体系中编译器的作用。通过本发明，可以将神经网络应用的开发和神经网络芯片的研发解耦合(decoupling)，硬件可以做得足够简单，致力于提高效率和集成度，同时又能支持任意的神经网络应用。

[0093] 本文中目标硬件为各种神经网络加速器和类脑计算芯片，这些芯片通常由若干处理核构成，每个处理核可以接受M个输入，与M×N的矩阵进行矩阵向量乘运算，得到的N个结果，经过硬件内置的激活函数或者内置的硬件神经元模型得到最终的N个输出。目标硬件由大量这样的处理核构成，处理核之间可以进行通信，本公开的硬件神经网络转换技术(图1中的编译层1200)仅仅要求处理核的每个输出可以发送到其他处理核的某个输入上即可。

[0094] 图1示出了根据本发明实施例的硬件神经网络转换技术的应用情境1000的示意图。

[0095] 如图1所示，本公开贡献在于在神经网络应用1100和神经网络芯片1300之间提供了一个编译层1200。编译层1200将神经网络应用转换成功能基本等效、同时又满足神经网络芯片的约束条件1400的网络，其表现为硬件神经网络的参数文件。基于该参数文件，后续可以利用某种映射算法来将硬件神经网络映射到神经网络硬件上去，使得映射后的神经网络硬件能够运行所述神经网络应用的功能。编译层1200所进行的转换对于应用开发人员是透明的。之所以称其为编译层，是因为它的功能和作用类似于编程领域中的将高级编程语言转换为二进制可执行程序(或汇编语言)的编译器，高级语言编程人员无需了解编译器的细节，只需进行高级语言编程，由编译器将高级语言编写的程序转换成计算机硬件能够理解和执行的二进制可执行程序(汇编语言)，在转换过程中编译器会考虑二进制可执行程序(汇编语言)的约束条件。

[0096] 图2示出了根据本发明实施例的编译层1200执行的硬件神经网络转换方法200的总体流程图，硬件神经网络转换方法200将神经网络应用转换为满足硬件约束条件的硬件神经网络。

[0097] 硬件神经网络转换方法200包括神经网络连接图获得步骤S210、神经网络连接图拆分步骤S220、神经网络基本单元转换步骤S230和基本单元硬件网络连接步骤S240。

[0098] 在步骤S210中，执行神经网络连接图获得，获得神经网络应用对应的神经网络连接图，神经网络连接图是一个有向图，图中的每个节点表示一层神经元，每条边表示层间的连接关系。

[0099] 大多数多层感知机或者简单的卷积神经网络表示成这种图的形式通常是一条简单的链状结构，复杂的神经网络可以是任意形式的图。

[0100] 通常，从神经网络模型文件解析得到神经网络连接图。不过并非只能从模型文件里面读取解析出一个神经网络连接图，也可能有如下情况，例如某些神经网络软件模拟器，通过几行代码就可以在运行时构建一个神经网络连接图。

[0101] 图3给出了神经网络连接图的示例300。其中节点1、2、3、4、5的每个表示为一层神经元。

[0102] 后续将以此图3所示的神经网络连接图为例给出编译层1200的硬件神经网络转换方法的具体示例。关于示例中涉及的硬件的基本模块的约束，示例性配置如下：硬件的基本模块能处理16×16规模的矩阵加权操作，每个基本模块上仅有32个8位宽的寄存器，矩阵的16×16＝256个参数仅仅记录了索引值，输入输出数据位宽为6bit，然后进行ReLU激活函数的运算产生输出，硬件仅支持1对1的通信，即硬件基本模块的16个输出，每个输出结果仅能发送给其他任意一个模块的一个输入。

[0103] 在示例中，图3中所示的神经网络连接图各节点和边的详细情况如下：

[0104] 其中节点1为6×6的图像，共36个神经元。

[0105] 边1-2为卷积操作，卷积核大小为3×3，共8个卷积核，因此节点2为8×6×6共288个神经元，激活函数为ReLU。

[0106] 边1-3为maxpooling，pooling范围是2×2，因此节点3为3×3共9个神经元。

[0107] 边3-5为全连接，节点5有5个神经元，激活函数为ReLU。

[0108] 节点4有32个神经元，边2-4和边3-4均为全相联，激活函数为Sigmoid。

[0109] 这里的神经网络连接图给了神经网络应用一个通用的描述，便于拆分成多个神经网络基本单元。

[0110] 在步骤S220中，进行神经网络连接图拆分，将神经网络连接图拆分为神经网络基本单元，每个神经网络基本单元中，只有入节点和出节点，不存在中间层节点，入节点和出节点之间全相联，，而且入节点中的神经元的所有出度在该基本单元内，出节点中的每个神经元的所有入度在该基本单元内。

[0111] 图4给出了神经网络基本单元400的示例性示意图。

[0112] 神经网络基本单元400包括两个入节点I1和I2、三个出节点O1、O2和O3，这里的每个节点均表示原始神经网络应用中的一层神经元。可见，基本单元400中不包括中间层节点，切入节点与出节点之间全相联，即入节点I1连接到出节点O1、O2和O3的每个，入节点I2也连接到出节点O1、O2和O3的每个。

[0113] 神经网络连接图拆分算法主要包括两步：

[0114] (1)将连接图中所有的节点，根据其前向顶点集合进行分组，同一组的顶点具有相同的前向顶点集合；

[0115] (2)若一个顶点的后向顶点分布在多个分组中，则增加若干复制顶点，每个复制顶点与其中一个分组相连。

[0116] 此时每个分组与其共同前向顶点集合构成了一个神经网络基本单元，所有复制顶点与其源节点也构成了一个神经网络基本单元。此时整个神经网络连接图被分解为了若干神经网络基本单元。

[0117] 下面以图3所示的神经网络连接图为例，结合图5(a)-(c)来说明将神经网络连接图拆分为多个神经网络基本单元的过程示例。

[0118] 图5(a)即为图3所示的神经网络连接图。

[0119] 首先根据前驱顶点集合进行分组，节点2的前驱为节点1，节点3的前驱为节点1，因此节点2和3划分为一组，记为组23；节点4的前驱为2和3，节点5的前驱为3，因此节点4单独为1组，记为组4；节点5也单独为一组，记为组5。图5(b)以各个节点的颜色来示出了节点的分组情况。

[0120] 此时图5(b)中节点3的后向节点包含了组4和组5，因此增加两个节点3′和3″，其大小与节点3完全相同，节点3与这两个顶点之间的连接即对应神经元以权重1相连，完全复制节点3，而节点3′连接节点4，节点3″连接节点5，连接方式与原节点3与节点4和节点5的连接方式相同。此时节点3与两个复制节点也构成了一个基本单元，记为组33。

[0121] 此时将网络划分为4个基本单元，如图5(c)中以4种不同颜色的边所示出了4个基本单元，具体地，节点(1,2,3)构成一个基本单元，节点(3,3′,3″)构成一个基本单元，节点(2,3′,4)构成一个基本单元，节点(3″,4)构成一个基本单元。

[0122] 回到图2，在神经网络连接图拆分步骤S220完成之后，前进到步骤S230。

[0123] 在步骤S230中，进行神经网络基本单元转换，将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络，称之为基本单元硬件网络，一个神经网络基本单元对应于一个或多个神经网络硬件的基本模块虚拟体，每个神经网络硬件的基本模块虚拟体均满足神经网络硬件的基本模块的连接度约束条件，且能够直接映射到神经网络硬件的基本模块。

[0124] 在一个示例中，神经网络基本单元转换步骤包括：对每个神经网络基本单元重建网络拓扑；以及针对重建的网络拓扑，进行权重参数确定。

[0125] 如前所述，神经网络硬件芯片上的硬件处理核通常经过了简化，能力往往比相同规模的神经网络应用要弱。上述重建拓扑旨在改变拓扑增强硬件网络能力；进行权重参数确定旨在微调权重逼近原神经网络网络应用的输出。

[0126] 后续将参考附图6，详细描述基本单元转换重建网络拓扑操作和权重参数微调操作。

[0127] 需要说明的是，步骤S230的转换是针对各个神经网络基本单元分别进行的。

[0128] 在一个优选示例中，按照神经网络连接图的拓扑序，来逐个转换各个神经网络基本单元。这样做是基于如下考虑：基本神经网络单元所进行的计算相对简单，因此微调可以很快收敛，但仍然会有少量误差。若误差逐层积累，最终整个神经网络的误差会变得很大。

因此各个神经网络基本单元并不相互独立、并行地进行上述转换算法，而是各个神经网络基本单元按照拓扑序，逐个进行转换。转换过程中的重新训练所需要的训练数据来源如下：

[0129] (1)输入数据：由于按照拓扑排序进行训练，当进行某神经网络基本单元的转换时，该当前神经网络基本单元之前的所有神经网络基本单元应当都已经完成了转换，因此该当前神经网络基本单元训练所用的输入数据是由原网络的训练样本经过前面这些已经转换之后的神经网络基本单元的计算产生的输出，由此可以将前面神经网络基本单元的转换误差代入到这一层的微调中以尝试消除；

[0130] (2)输出数据：输出数据仍然为原始网络对应神经元在对应样本下的输出值。

[0131] 以链状的神经网络连接图为例，所有样本在原神经网络中在各层的输出值为{Y_1,Y_2,…,Y_N}，通过Y_1和Y_2作为输入和输出数据，训练第一个神经网络基本单元$f_1(Y)$，使得其输出值$Y_2'=f_1(Y_1)$与Y_2的误差尽可能小，接下来以Y_2'和Y_3作为输入和输出数据，训练第二个神经网络基本基本单元$f_2(Y)$，使得其输出值$Y_3'=f_2(Y_2')$与Y_3的误差尽可能小，逐个转换和微调，直到最后一层。

[0132] 通过这种方式可以避免误差的逐层积累，使得最终得到的神经网络与原网络的误差尽可能小。

[0133] 在神经网络连接图为有向无环图的情况下，可以直接按照神经网络连接图的拓扑序，逐个转换各个神经网络基本单元；

[0134] 在神经网络连接图为有环有向图的情况下，例如，对于RNN，首先将有环有向图的环拆开，使得神经网络连接图变成有向无环图，然后按照有向无环图的拓扑序，逐个转换各个神经网络基本单元。

[0135] 按照所述拓扑序，进行转换后的各个神经网络基本单元的训练，其中重新训练所需要的训练数据来源为：训练输入数据是训练样本在经过拓扑序在前的基本单元硬件网络之后产生的输出，训练输出数据是训练样本在原神经网络应用对应层产生的输出。

[0136] 经过上述对每个神经网络基本单元的转换操作之后，每个神经网络基本单元被转换为基本单元硬件网络，既确定了基本单元硬件网络中的各个基本模块虚拟体之间的连接，也确定了有关权重参数等配置。

[0137] 例如，仍以前述示例进行说明，对于图5(c)所示的各个组(神经网络基本单元)，按照拓扑序，先转换组23，然后是组33，最后是组4和组5。

[0138] 在神经网络基本单元转换步骤S230完成之后，前进到步骤S240。

[0139] 在步骤S240中，进行基本单元硬件网络连接，将得到的基本单元硬件网络按照拆分的顺序连接起来，生成硬件神经网络的参数文件。

[0140] 当所有的神经网络基本单元均完成转换之后，根据拆分重新将转换之后的各个基本单元链接起来，由于每个基本单元均被转换成了一堆虚拟核之间组成的小网络，链接之后得到的是虚拟核组成的硬件神经网络。这里的虚拟核即前文所述的基本模块虚拟体。

[0141] 然后再根据硬件的物理网络拓扑特点，使用相应的映射算法，将虚拟核映射到物理网络上，以实现高效的通信。

[0142] 另外，如果目标硬件的处理核支持时分复用，可以综合考虑通信和权重复用的特点，将权重值相同的虚拟核或连接紧密的虚拟核映射到同一个物理核。

[0143] 如前所述，现有的技术路线均是让神经网络的应用和芯片直接适配，要么将芯片

直接去适配应用的自由度，这会带来性能瓶颈；要么将芯片的约束暴露给应用，这约束了应用的能力。相对比，本发明实施例的硬件神经网络转换方法，在神经网络应用和神经网络芯片之间加上了一个中间层，通过一种相当于传统计算机体系当中的编译的技术解决了神经网络应用与神经网络应用芯片之间的适配问题，同时解耦合了应用和芯片的开发。

[0144] 另外，本发明实施例的硬件神经网络转换方法，对于任意的复杂神经网络和满足硬件抽象的任意硬件，提供了一种通用的流程，可以将复杂神经网络转换成满足该硬件约束条件的特定网络，且功能上与原网络基本等效。该流程的核心在于将复杂网络进行分解，由于每个基本单元所做的运算相对简单，转换过程相比直接转换整个网络更由保障能收敛，且收敛速度也更快。

[0145] 而且，本发明实施例的硬件神经网络转换方法，通过对神经网络连接图中的节点进行分组，将神经网络拆分成若干基本单元，使得基本单元内任意一个节点的入边或出边全部在该基本单元内，从而在基本单元内解决了连接度的问题之后，将转换完的基本单元重新链接起来，得到的网络仍然能满足连接度的要求。

[0146] 另外，在前面所述的一个示例中，按照拓扑序逐个模块进行转换，将前面产生的误差引入到后面的微调中，使得各个基本模块转换引入的误差不会逐层积累。

[0147] 另外，在一个示例中，在神经网络应用具有卷积层的情况下，在神经网络连接图拆分步骤S220之前，可以对于神经网络应用的卷积层进行网络压缩，在本文中也称之为硬件无关优化，因为该优化是和神经网络硬件芯片没有关系的。

[0148] 硬件无关优化可以将神经网络的规模减小，对神经网络进行压缩。各种相关技术均可用于此处，例如现有技术基于行列式点过程(Determinantal Point Process,DPP)提取神经元多样性来进行网络压缩的技术，但该现有技术仅适用于简单的全连接网络，不能直接适用于常见的卷积神经网络。

[0149] 首先，概要介绍一下行列式点过程DPP。

[0150] DPP是一种获取多样性子集的技术，假设由N个元素组成的集合L，总共有2^N个子集。有一个N×N的矩阵K。若从这N个元素中采样出子集$A \subseteq L$的概率$P(A) \propto |K_A|$，其中K_A表示K由集合A中的元素所对应的行和列张成的子矩阵，$|K_A|$表示K_A的行列式，则称该过程为DPP。若矩阵元K_{ij}表示第i个元素和第j个元素的相似度，则子集中的元素相似性越低，DPP采样得到该子集的概率越高，因此概率最高的子集是多样性最高的子集。

[0151] 根据本发明的一个实施例，通过巧妙的设计将该现有技术DPP推广到更加实用的卷积神经网络中。

[0152] 具体地，在卷积神经网络中，每一层会有若干个特征图(feature map)，这些特征图所携带的信息通常是有冗余的。我们通过将这些特征图在所有样本上产生的输出之间的相似性作为K的矩阵元，利用DPP得到多样性最高的子集，保留该子集，丢弃掉其他特征图节点，将丢弃的特征图所对应的向量投影到保留的特征图所张成的线性空间中，用丢弃的特征图的投影长度与其原向量长度的比值作为加权系数，将丢弃的特征图与下一层神经元的连接权重加权累加到保留的特征图与下一层神经元的连接权重上。

[0153] 仍以前面的图3所示的连接图为例，说明对每层神经元进行硬件无关优化的方法。

[0154] 如前所述，图3中主要包括3种类型的连接，卷积、全连接和maxpooling，其中maxpooling是无参数层，无需优化，而其他两种层均可利用基于DPP的多样性检测来完成尺

寸的压缩。

[0155] 例如卷积操作边1-2，节点2包含了8个特征图，通过求出网络的训练样本在这8个特征图上产生的输出组成的向量Y_i之间的相似度来构建一个8×8的矩阵，通过DPP方法采样出多样性最高的子集，假设包含了6个特征图，设其输出向量分别为$Y_1,\cdots,Y_6$，则将剩下的Y_7在$Y_1,\cdots,Y_6$所张成的线性空间内投影，在$Y_1,\cdots,Y_6$上的投影值分别为$\alpha_1,\cdots,\alpha_6$，则边2-4的原先为8×6×6个神经元与32个神经元的全连接，将Y_7所对应的6×6个神经元与32个神经元的连接权值乘以α_i累加到Y_i所对应的6×6的神经元与32个神经元的连接权值上去。同理处理未被选中的Y_8，此时节点2的尺寸变为了6×6×6共216个神经元。

[0156] 节点4和5由于是输出节点，无法进行压缩，而节点3是maxpooling得到的节点，因此也无法进行压缩。所以通过硬件无关优化，将节点2变为6×6×6规模。

[0157] 利用本发明实施例的针对卷积神经网络的网络压缩算法，通过推广现有技术方案，利用DPP提取多样性子集的方法，选取出卷积神经网络的每一层中，多样性最高的特征图子集，丢弃剩下的特征图节点，以此有效减少卷积神经网络每一层的特征图数量，减少网络的规模，降低硬件的资源开销；并利用投影和微调的方式，降低对网络精度的影响。通过该方法，可以有效地去除网络中的冗余，减少对硬件资源的占用。

[0158] 下面结合附图6到9详细描述神经网络基本单元转换的具体实现示例。

[0159] 如前所述，神经网络基本单元转换230可以包括网络拓扑重建操作2310和权重参数微调2320，其中网络拓扑重建操作2310可以包括重编码2311、特殊函数处理2312和完全展开2313，权重参数微调2320可以包括参数初始化微调2321、权重取值范围微调2322、低精度权重微调2323。网络拓扑重建操作2310旨在增强硬件网络能力，权重参数微调2320旨在逼近原神经网络应用输出。

[0160] 下面针对各个具体操作进行详细说明。

[0161] 1、利用自编码器进行层间数据重编码2311

[0162] 由于神经网络硬件通信时传递的数据精度通常很低，如果直接将原网络的数据四舍五入，很有可能丢失信息。因此要将神经网络层间传递的数据用低精度重新编码，使得在低精度下仍然不损失主要信息。

[0163] 自编码器(autoencoder)是一种利用神经网络进行信息编码的技术，由3层神经元组成，包括输入层、隐藏层和输出层，其中输出层的节点数和输入层节点数相同。训练该网络，使得输出层的值与输入层的值尽可能相近。则隐藏层的值为输入数据的另一种编码，从输入层到隐藏层的计算为编码过程，对应于编码器，而从隐藏层到输出层的计算为解码过程，对应于解码器(参见图7)。由于隐藏层解码得到的数据与输入层接近，因此隐藏层的编码没有损失主要信息。

[0164] 图7示出了对于三层神经网络利用自编码器重新编码后得到的扩充后的三层神经网络。如图7所示，1)对于神经网络的每一层(层FC1、FC2和FC3，如标号1所示)的层间输出向量(图7所示的层FC1与FC2之间的输出向量，FC2与FC3之间的输出向量)，2)我们构建一个隐藏层使用硬件数据精度的自编码器(图7中所示的一组编码解码，如标号4所示)，且隐藏层的节点数量多于层间向量数据的维度，通过训练自编码器，得到了层间向量在硬件数据精度下的编码，注意自编码器的输入和输出仍然是原精度例如浮点精度，而只有中间的隐藏层用硬件精度。3)将自编码器插入到神经网络的层间，替换掉原来的层间向量，如标号2所

示。4)对于每一个连接,其输入节点的解码器、连接的权重矩阵和输出节点的编码器将合并成一个更大规模的连接矩阵,如标号3所示,相比于旧层FC1、FC2、FC3的规模,新层FC1'、FC2'、FC3'的规模扩大了。

[0165] 通过上述方式,将神经网络的层间向量更换成了硬件精度编码的向量,确保信息不会由于层间向量所使用的精度而丢失,同时扩大了连接矩阵的规模,增大的硬件网络的逼近能力。

[0166] 下面说明一下卷积层的自编码器的处理示例。例如对c×w×h的一层(c个通道,w宽,h高),卷积核为k×k,得到的隐藏层为c′×w×h,经过激活函数,再经过k×k卷积核和激活函数,解码回c×w×h。此时编码器和解码器都是卷积操作。

[0167] 如果后面连接的还是卷积层,则从当前层的隐藏层到下一层的隐藏层相当于连续进行了3次卷积操作,先是解码器,然后是卷积层,然后是编码器,连续3个卷积操作可以合并成一个卷积操作,例如连续3个3×3的卷积操作,可以合并成一个7×7的卷积操作,因为每个像素与前面3×3领域内的像素连接,而这个3×3的领域内的像素与再前面一层5×5范围内的像素相连,再往前是7×7,这个7×7的卷积核可以通过这3个3×3的卷积核初始化。

[0168] 如果后面连接的是全相联的层,则直接把解码器的卷积操作展开成矩阵,然后和后面的全相联矩阵以及后面一层的编码器矩阵相乘,得到的结果用来初始化隐藏层之间的大矩阵。

[0169] 仍以前述的神经网络应用为例,说明重编码过程,对于图5(c)所示的各个组,以组23为例,输入图像为6×6,由于直接四舍五入到6bit可能会丢失图像中的重要信息,因此可以将输入图像重编码,设置一个隐藏层为2×6×6的自编码器,隐藏层的输出精度为6bit,得到编码器和解码器,网络的输入图像首先通过编码器处理之后输入到网络中,节点1变为2×6×6的规模,相比于原节点1的6×6规模,可见重编码后规模扩大了。

[0170] 同理处理节点2,假设通过上述方式,节点2变为9×6×6,相比于原节点2的8×6×6的规模,可见重编码后规模扩大了。

[0171] 节点3为maxpooling得到的结果,因此无需重编码,但由于输入层重编码为2×6×6,因此节点3也相应变为2×3×3,共18个神经元。

[0172] 在另一个示例中,如下配置自编码器,自编码器的输入是层间输出经过激活函数之前的,输出是经过激活函数之后的。例如,自编码器的输入是FC1未经过FC1激活函数的输出结果,输出是FC1经过了激活函数的输出结果。相当于用自编码器学习了一下FC1的激活函数(标准的自编码器是直接输入和输出相同的)。FC2的输出同样处理。换句话说,原网络形式为:FC1矩阵向量乘输出->FC1激活函数->FC2矩阵向量乘输出->FC2激活函数->...。现在将各个激活函数替换成对应的自编码器,FC1矩阵向量乘输出->FC1编码器->FC1解码器->FC2矩阵向量乘输出->FC2编码器->FC2解码器->...,其中FC1解码器->FC2矩阵向量乘输出->FC2编码器会合并成一个大矩阵。这样同样达到了下述效果:将神经网络的层间向量更换成了硬件精度编码的向量,确保信息不会由于层间向量所使用的精度而丢失,同时扩大了连接矩阵的规模,增大的硬件网络的逼近能力。

[0173] 2、特殊函数处理2312

[0174] 由于神经网络中通常不仅有矩阵乘、卷积等操作,还有一些特殊的操作,例如卷积神经网络中非常常用的maxpooling操作。其核心是max函数,这些函数通常没有参数,都是

固定的计算,因此可以为这些函数构造专门的神经网络实现其功能。

[0175] 例如由于max函数可以用多个ReLU激活函数(ReLU(x)＝max(x,0))来实现:

[0176] max(a,b)＝0.5ReLU(a+b)+0.5ReLU(a-b)+0.5ReLU(b-a)

[0177] +0.5ReLU(-b-a)

[0178] 因此,max操作可以用如图8所示的神经网络替换。

[0179] 前述示例中的节点3需要进行特殊函数处理。边1-3是对节点1中每4个神经元的输出进行求最大值操作得到节点3中的一个输出,共18个这样的操作。

[0180] 图8中所示的神经网络可以求两个输入值得最大值,通过3个这样的网络组合,我们可以求出4个输入值的最大值,即两两求最大,再进一步求两个最大值的最大值。再用18个4输入求最大值的网络替换掉边1-3的maxpooling。

[0181] 当然,如果硬件资源中本来就有特殊函数的计算资源,相应的特殊函数处理也可以省去,直接使用硬件提供的计算资源。

[0182] 3、完全展开2313

[0183] 由于目标硬件仅仅支持固定规模的矩阵向量乘操作,对连接度有约束,在神经网络基本单元的规模超过硬件的约束的情况下,需要对神经网络基本单元中的大规模矩阵乘法(可选地,连同卷积操作)进行分解、合并,本文中将之称为完全展开操作,经过完全展开,神经网络基本单元被分解为了基本模块虚拟体(或称之为虚拟核)之间的相互连接。

[0184] 图9示出了根据本发明一个实施例的对于大规模矩阵乘法操作的完全展开2313的示例性示意图。

[0185] 图9中,M和N限定了虚拟核能够处理的矩阵大小,A和B限定了实际的大规模矩阵相对于虚拟核的矩阵大小的规模,图中为表示方便,假设M＝N,A＝B＝2。

[0186] 如图9所示,对于大规模的矩阵乘法和卷积操作,该实施例使用3组虚拟核来进行相关计算:(1)其中计算组23132负责真正的运算,将大规模的矩阵乘法(图9中的连接矩阵为M*A)划分成多个小矩阵(图9中为4个M*N的矩阵),分布在这一组虚拟核中进行真正的计算,每个虚拟核负责一个小矩阵运算,对于大规模的卷积,则将卷积条带化,同样分解成多个小矩阵进行处理;(2)另外两组虚拟核,多播组23131和归约组23133,分别用作多播和归约,用于多播的虚拟核的每个将每个输入数据复制多份(图9中示出为两份)分发到需要该数据的小矩阵中,虚拟核的输出为N维向量,而多播操作为一变二操作,因此多播操作的虚拟核的输入为N/2,也即M*A/4,计算组23132中的每个虚拟核接收来自两个执行多播操作的虚拟核的输出,即形成了M维(此例中也即N维)输入,然后计算组23132中的每个虚拟核执行M维向量与M*N维矩阵的矩阵向量乘操作,得到的结果为N维向量,将该N维向量分为两半,分别输出到两个执行归约的虚拟核,执行归约的虚拟核,用于归约的虚拟核将各个小矩阵对同一个神经元的输出数据累加起来,得到最终的输出,图9中示出为N*B个位的输出。

[0187] 图9所示的示例以M＝N的虚拟核以及A＝B＝2的实际神经网络基本单元规模来对神经网络基本单元的完全展开操作进行了说明,需要说明的是,此仅为示例,而不应作为对本发明的限制,如果M和N不相等,则可以根据M和N实际大小分配多播层和归约层的核数量。

[0188] 经过完全展开,神经网络基本单元被分解为了一系列虚拟核之间的相互连接,每个虚拟核均满足硬件处理核的连接度约束条件。

[0189] 仍以前面的示例为例,来说明基本单元的完全展开操作,此时边2-3的输入为2×6

×6,输出是9×6×6,通过3×3的卷积,输出的9张特征图中任意一个坐标x,y所对应的9个点,其输入来自于输入的2个特征图中对应位置周围的3×3范围的共18个点,因此存在18×9这样的全连接结构,卷积操作可以转化为6×6共36个规模不超过18×9的全连姐操作(图像的边缘可能不足18个输入节点,因此这里说的是规模不超过)。但18×9的规模仍然超过了硬件16×16的约束限制,因此拆分成2个9×9的小矩阵乘操作,而对于节点1中的每个输出,需要为3×3=9个18×9的矩阵提供数据,而因为每个又拆分为了2个9×9的小矩阵,因此需要为18个9×9的小矩阵提供输入数据,同时,节点1中的每个输出,在边1-3中也需要提供1个数据,因此节点1中的每个输出需要将数据发送给19个硬件基本模块,而硬件的规模为16×16,在完全展开的过程中,节点1中的每个输出先发送给1个硬件基本模块,得到16份输出,其中15个直接连接到需要该数据的15个硬件基本模块上,最后一个连接到一个硬件模块上再复制出4个输出,并连接到剩下的4个需要的该数据的硬件基本模块上。由此,神经网络基本单元被分解为了一系列虚拟核之间的相互连接,每个虚拟核均满足硬件处理核的连接度约束条件。

[0190] 4、权重参数微调2320

[0191] 接下来最终网络拓扑重建2310步骤后得到的基本单元硬件网络的权重参数。对于基本单元硬件网络的权重参数,可以首先根据原网络权重参数进行初始化,之后逐步将权重的约束引入,每次均对网络参数进行微调,使得硬件网络与原网络的误差尽可能减小。

[0192] 为便于理解,在详述如何进行权重参数微调之前,首先介绍根据本发明实施例对于硬件权重取值的抽象操作。很多硬件通常会对权重进行很大的简化,例如有些硬件使用8位整型存储权重,有些硬件使用动态定点数存储权重(即小数点位置可以配置的定点数),IBMTrueNorth每个神经元分配了3个8位整型寄存器,其所有权重从这3个整数和0当中选取。针对各种硬件设计,可以将硬件权重取值的约束作如下抽象。

[0193] 权重矩阵W取值范围可看作一个集合S^P,集合中每个元素都是关于参数P的函数,其中P为硬件可以配置的参数。例如:

[0194] 对于使用8位整型的硬件,无参数,集合S={-128,127,…,-1,0,1,…,127};

[0195] 对于动态定点数,参数P为小数点位置,集合$S^P=\left\{-\frac{2^N}{2^P},-\frac{2^N-1}{2^P},\ldots,-\frac{1}{2^P},0,\frac{1}{2^P},\ldots,\frac{2^N-1}{2^P}\right\}$;

[0196] 对于IBM TrueNorth,参数P为寄存器的取值,集合$S^{P_1,P_2,P_3}=\{0,P_1,P_2,P_3\}$

[0197] 而权重矩阵中的每个元素W_{ij}可以独立的从S^P中选择,即可以独立配置索引k_{ij},使得$W_{ij}=S^P_{k_{ij}}$,因此权重矩阵可以配置的是集合参数P和各个权重在集合中取值的索引k_{ij}。

[0198] 在给出硬件权重取值约束的抽象之后,下面介绍根据本发明实施例的权重参数确定方法示例。

[0199] 首先,根据原神经网络的权重初始化构造出的基本单元硬件网络的权重。并进行权重参数的微调,使得权重满足硬件的权重约束。主要分为以下3个步骤。

[0200] (1)首先使用浮点精度表示权重,对构造出的网络进行重新训练,使得与原网络的误差尽可能小,以此弥补硬件激活函数或硬件神经元模型与原神经网络之间的差异。此步骤对应于图6中的参数初始化微调2321操作。

[0201] (2)根据第(1)步训练得到的参数，利用EM((Expectation Maximization，期望最大化)算法确定一个最好的P(P为上述硬件权重约束抽象中所提及的可配参数)和k_{ij}(即上述各个矩阵元在集合S^P中取值的索引)，此时所有的权重参数均可以表示为P的函数，重新训练以调节P。此步骤对应于图6中的权重取值范围微调2322操作。

[0202] EM是算法是为了选择合适的P，使得浮点精度的权重参数四舍五入到S^P集合中之后，引入的误差尽可能小，即最小化目标函数$J(k,P)=\sum\left(W_{ij}-S^P_{k_{ij}}\right)^2$。按照标准的EM算法：

[0203] E-step:固定P＝$P^{(t)}$，令$k_{ij}^{(t)}=\arg\min J\left(k\middle|P^{(t)}\right)$

[0204] M-step:固定$k_{ij}=k_{ij}^{(t)}$，令$P^{(t+1)}$＝arg minJ(P|$P^{(t)}$)

[0205] 该算法在IBMTrueNorth那种共享权重的情况下，会自动退化成k-means算法，通过计算权值分布的k个重心，从而将寄存器的值设置为这些重心的值，让所有权值的索引设置为距离最近的重心。

[0206] (3)固定第(2)步训练得到的P，将所有权重初始化为对应的$S^P_{k_{ij}}$，重新训练以调节k_{ij}，所有权重仍然使用浮点精度存储，但在训练的前馈过程中，所有的权重参数四舍五入到S^P中最接近的值，然后带入前馈计算，而在反馈和更新权重时，仍然使用浮点精度，更新浮点精度的权重值。此步骤对应于图6中的低精度权重微调2323操作。

[0207] 仍以前述示例为例说明权重微调的过程，对于组23，首先是使用浮点精度对权重进行微调，弥补由于自编码器等操作引入的误差。

[0208] 然后根据所得到的参数，对每个硬件基本模块内的256个参数运行k-means算法，聚合到32个类中，每个参数均用类的重心表示。并进行第二次微调，调节各个模块的32个重心的数值。

[0209] 最后将训练得到的重心值填进32个寄存器中，进行第三次微调，此时所有的权重参数用浮点值表示，在前馈的过程中，找到该浮点值最近的重心值带入计算，反馈得到的梯度用来更新权重的浮点值，经过微调，确定各个权重参数的索引值。

[0210] 至此，组23完成转换，将训练数据经过转换之后的组23得到节点2和节点3的输出值，用这些输出值作为后续的组33转换过程中用到的训练数据。逐个完成组33、组4和组5的转换。

[0211] 前面参考附图并结合示例详述了根据本发明实施例的硬件神经网络转换方法以及其中的各个步骤的具体实现。需要说明的是，这些过程中的详细示例是为了本领域技术人员透彻理解而给出的，不应将这些详细示例理解为对本发明的限制。本发明的具体实现可以根据需要进行各种变化。

[0212] 例如，在先前示例中，在对目标硬件的抽象中，硬件所支持的通信要求是每个处理核的输出，只能拥有一个目标节点，因此约束了神经网络中每个神经元的出度。针对此约束，在图2所示的将神经网络连接图拆分步骤中，通过增加复制节点来增大出度，在神经网络基本单元转换步骤中的完全展开的操作中，使用了一组虚拟核用作多播。不过，显然，如果硬件本身支持一对多的通信模式，这些额外的复制节点和用于多播的处理核均可以省

去,以减少对硬件资源的开销。

[0213] 另外,前述很多步骤是针对一定的硬件约束而制定的,如果目标硬件不存在相应的约束,则相应的流程可以省略。例如权重微调的第2步,通过EM算法和重新训练来确定参数P,对于使用固定精度而不存在参数P的硬件,这样的步骤可以省略。而对于权重微调的第3步,主要是针对权重精度低而设计的,如果目标硬件本身支持的是浮点精度的权重,相应的步骤也可以省略。

[0214] 此外,在前述示例中,使用了特殊函数的处理,使得在硬件不支持特殊函数处理的情况下也能顺利完成特殊函数的计算,但如果硬件资源中本来就有特殊函数的计算资源,相应的处理也可以省去,直接使用硬件提供的计算资源。

[0215] 此外,如果硬件提供了额外的加法器,可以将不同处理核输出累加到一起,在完全展开策略中用于做归约的处理核也可以省去,而直接使用硬件提供的加法器来完成相应操作。

[0216] 进一步,需要说明的是,本发明实施例的硬件神经网络转换技术是普适性的,适用于各种神经网络,ANN(artificial neural network,人工神经网络)、SNN(Spiking Neuron Networks,脉冲神经网络)和RNN(Recurrent Neural Networks,循环神经网络)等。

[0217] 前面技术细节中,主要面向ANN形式的神经网络进行了讨论,对于SNN和RNN,本发明实施例的技术方案同样适用。

[0218] 1、SNN的处理

[0219] 若原神经网络网络应用为SNN:由于SNN中普遍使用频率编码,即以神经元发放电脉冲的频率来表示其所传递的数据,因此对原始神经网络应用以稳定频率的电脉冲作为输入,记录各个神经元的电脉冲发放频率,以此作为神经网络基本单元转换步骤S230中使用的训练数据。

[0220] 若神经网络硬件芯片涉及的模型为SNN:对于SNN的神经元模型,对于稳定的电流输入,神经元通常会产生稳定频率的电脉冲发放,而对突触输入稳定频率的电脉冲,突触也会产生稳定的电流输入到神经元当中。这两个关系通常都连续且可导,因此可以进行梯度的计算,因此可以使用反向传播算法进行训练。

[0221] 2、RNN的处理

[0222] RNN的神经网络连接图是有环图。

[0223] 如前所述,对于各个神经网络基本单元的转换,优选按照拓扑序进行。拓扑序转换要求神经网络的连接图能够进行拓扑排序,而仅有有向无环图可以进行拓扑排序,对于有环存在的神经网络,可以将存在的环拆开,使得神经网络连接图变成有向无环图,此时可以进行上述转换,转换完之后将环重新拼接起来,并对整个网络进行整体微调,使其逼近原网络。

[0224] 在另一实施例中,本发明实现为硬件产品,例如编译器硬件,或者其他计算装置形式,其接收神经网络应用和/或神经网络连接图作为输入,还接收神经网络硬件芯片的配置(如约束等)作为输入,然后得到硬件神经网络的参数文件。基于该参数文件,利用某种映射算法来配置神经网络硬件芯片,神经网络硬件芯片就能够实现神经网络应用了。本发明实施例的计算装置用于将神经网络应用转换为满足硬件约束条件的硬件神经网络,包括存储器和处理器,存储器中存储有计算机可执行指令,当处理器执行所述计算机可执行指令时,

执行前述的硬件神经网络转换方法，该方法包括：神经网络连接图获得步骤，获得神经网络应用对应的神经网络连接图，神经网络连接图是一个有向图，图中的每个节点表示一层神经元，每条边表示层间的连接关系；神经网络连接图拆分步骤，将神经网络连接图拆分为神经网络基本单元，每个神经网络基本单元中，只有入节点和出节点，不存在中间层节点，入节点和出节点之间全相联，，而且入节点中的神经元的所有出度在该基本单元内，出节点中的每个神经元的所有入度在该基本单元内；神经网络基本单元转换步骤，将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络，称之为基本单元硬件网络，一个神经网络基本单元对应于一个或多个神经网络硬件的基本模块虚拟体，每个神经网络硬件的基本模块虚拟体均满足神经网络硬件的基本模块的连接度约束条件，且能够直接映射到神经网络硬件的基本模块；基本单元硬件网络连接步骤，将得到的基本单元硬件网络按照拆分的顺序连接起来，生成硬件神经网络的参数文件。有关神经网络连接图获得步骤、神经网络连接图拆分步骤、神经网络基本单元转换步骤、基本单元硬件网络连接步骤的功能和具体实现可以参考前面结合图2-9所做的描述，这里不再赘述。

[0225] 根据本发明的另一方面，提供了一种将神经网络软件应用编译为硬件神经网络的编译方法，可以包括：获得神经网络软件应用和神经网络硬件芯片的配置情况；基于神经网络硬件的配置情况，将神经网络软件应用转换硬件神经网络，所述硬件神经网络由神经网络硬件芯片的基本模块连接而成；输出硬件神经网络的参数文件，所述参数文件描述所述基本模块之间的连接关系以及各个基本模块的参数配置情况。

[0226] 根据本发明的再一方面，提供了一种神经网络软硬件协作系统，可以包括：神经网络硬件芯片，神经网络硬件芯片上具有基本模块，基本模块以硬件形式执行矩阵向量乘和激活函数的操作，神经网络硬件芯片上的基本模块的参数和基本模块之间的连接能够由确定格式的配置文件配置；编译层单元，用于将神经网络应用编译为硬件神经网络的参数文件，基于参数文件能够将硬件神经网络映射到一个或多个神经网络硬件芯片，映射后的一个或多个神经网络硬件芯片能够运行所述神经网络应用的功能。

[0227] 根据所述实施例的神经网络软硬件协作系统，所述编译层单元配置为执行下述方法：硬件配置数据获得步骤，获得神经网络硬件芯片的配置情况数据；神经网络连接图获得步骤，获得神经网络应用对应的神经网络连接图，神经网络连接图是一个有向图，图中的每个节点表示一层神经元，每条边表示层间的连接关系；神经网络连接图拆分步骤，将神经网络连接图拆分为神经网络基本单元，每个神经网络基本单元中，只有入节点和出节点，不存在中间层节点，入节点和出节点之间全相联，而且入节点中的神经元的所有出度在该基本单元内，出节点中的每个神经元的所有入度在该基本单元内；神经网络基本单元转换步骤，将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络，称之为基本单元硬件网络，一个神经网络基本单元对应于一个或多个神经网络硬件的基本模块虚拟体，每个神经网络硬件的基本模块虚拟体均满足神经网络硬件的基本模块的连接度约束条件，且能够直接映射到神经网络硬件的基本模块；基本单元硬件网络连接步骤，将得到的基本单元硬件网络按照拆分的顺序连接起来，生成硬件神经网络的参数文件。有关神经网络连接图获得步骤、神经网络连接图拆分步骤、神经网络基本单元转换步骤、基本单元硬件网络连接步骤的功能和具体实现可以参考前面结合图2-9所做的描述，这里不再赘述。

[0228] 本发明的硬件神经网络转换方法、计算装置、将神经网络软件应用编译为硬件神经网络的编译方法、神经网络软硬件协作系统做出了开创性的贡献，具有突出的技术效果。

[0229] 本发明提出了一种全新的神经网络和类脑计算的软硬件体系，通过在神经网络应用和神经网络芯片之间加上了一个中间编译层，解决了神经网络应用与神经网络硬件之间难以适配的鸿沟，既不需要限制神经网络应用本身的自由度和灵活性，也避免了硬件为实现自由度带来的性能瓶颈。

[0230] 同时，本发明将神经网络应用和芯片解耦合，神经网络应用无需针对不同的底层硬件重新开发，通过本发明，可以将一个训练好的神经网络适配到任意的神经网络芯片上。同时也提高了神经网络芯片的通用性，神经网络芯片的研发也无需增加新的结构便可支持应用中出现的新的特性。

[0231] 此外，本发明的技术方案的转换时间也远小于重新训练整个神经网络的时间，相比针对硬件重新设计和训练神经网络，效率要高很多。

[0232] 本公开的各个实施例提供了开创性的技术方案：

[0233] (1)提出了一种全新的神经网络和类脑计算的软硬件体系

[0234] 现有的技术路线均是让神经网络的应用和芯片直接适配，要么将芯片直接去适配应用的自由度，这会带来性能瓶颈；要么将芯片的约束暴露给应用，这将约束了应用的能力。本发明在应用和芯片之间加上了一个中间层，通过一种相当于传统计算机体系当中的编译的技术解决了该问题，同时解耦合了应用和芯片的开发。

[0235] (2)提出了一种神经网络应用的转换(编译)算法流程

[0236] 对于任意的复杂神经网络，和满足硬件抽象的任意硬件，本文提出了一种通用的流程，可以将复杂神经网络转换成满足该硬件约束条件的特定网络，且功能上与原网络基本等效。该流程的核心在于将复杂网络进行分解，由于每个基本单元所做的运算相对简单，转换过程相比直接转换整个网络更由保障能收敛，且收敛速度也更快。同时按照拓扑序逐个模块进行转换，将前面产生的误差引入到后面的微调中，使得各个基本模块转换引入的误差不会逐层积累。

[0237] (3)提出了一种通用的神经网络的拆分算法

[0238] 通过对神经网络连接图中的节点进行分组，将神经网络拆分成若干基本单元，使得基本单元内任意一个节点的入边或出边全部在该基本单元内，从而在基本单元内解决了连接度的问题之后，将转换完的基本单元重新链接起来，得到的网络仍然能满足连接度的要求。

[0239] (4)提出了一种针对卷积神经网络的网络压缩算法

[0240] 在一个具体实施例中，通过推广现有技术方案，利用DPP提取多样性子集的方法，选取出卷积神经网络的每一层中，多样性最高的特征图子集，丢弃剩下的特征图节点，以此减少网络的规模。并利用投影和微调的方式，降低对网络精度的影响。通过该方法，可以有效地去除网络中的冗余，减少对硬件资源的占用。

[0241] (5)提出了一种通用的神经网络转换算法

[0242] 根据一个具体实施例，通过拓扑重建，构建一个拓扑更复杂，能力更强的硬件神经网络。其技术核心包括通过自编码器实现的硬件精度编码，以解决硬件精度约束；特殊函数的处理，以解决硬件激活函数或神经元模型的约束；完全展开，以解决硬件连接度的约束。

[0243] 进一步地，在一个具体实施例中，通过多次权重微调，使得硬件神经网络逼近原神经网络的功能。其核心技术包括基于EM算法和低精度训练方法的权重设置。

[0244] 本公开的技术是通用的申请网络转换算法，适用于ANN、SNN和RNN等各种神经网络的处理。

[0245] 需要说明的是，附图中按某顺序显示了各个步骤，并不表示这些步骤只能按照显示或者描述的顺序执行，只要不存在逻辑矛盾，步骤执行顺序可以不同于所显示的。

[0246] 以上已经描述了本发明的各实施例，上述说明是示例性的，并非穷尽性的，并且也不限于所披露的各实施例。在不偏离所说明的各实施例的范围和精神的情况下，对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。因此，本发明的保护范围应该以权利要求的保护范围为准。

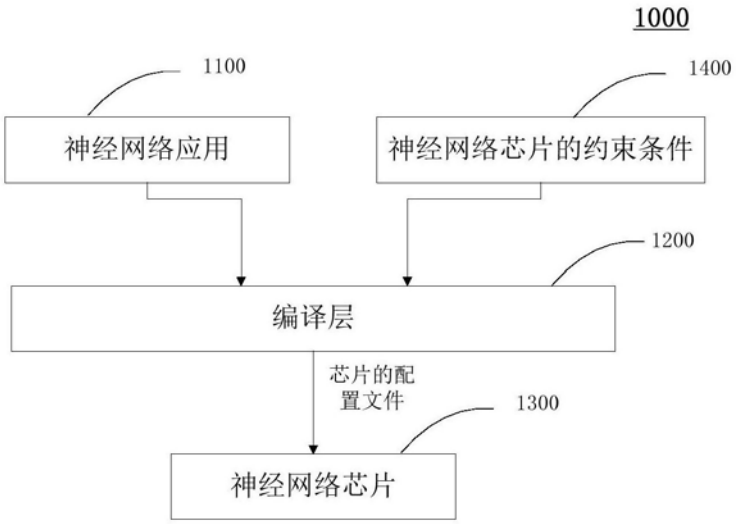

1000
1100
1400
神经网络应用
神经网络芯片的约束条件
1200
编译层
芯片的配置文件
1300
神经网络芯片

图1

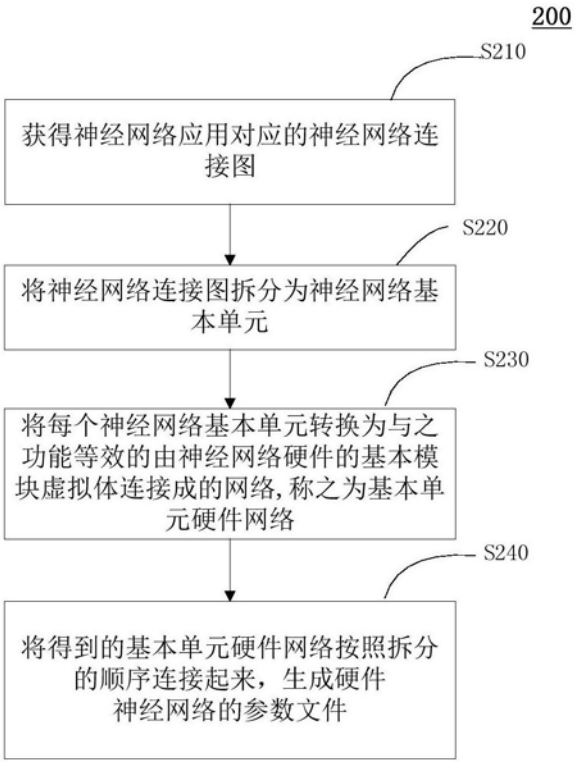

200
S210
获得神经网络应用对应的神经网络连接图
S220
将神经网络连接图拆分为神经网络基本单元
S230
将每个神经网络基本单元转换为与之功能等效的由神经网络硬件的基本模块虚拟体连接成的网络,称之为基本单元硬件网络
S240
将得到的基本单元硬件网络按照拆分的顺序连接起来，生成硬件神经网络的参数文件

图2

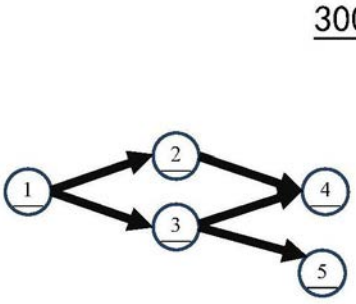


图3

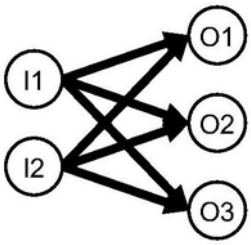
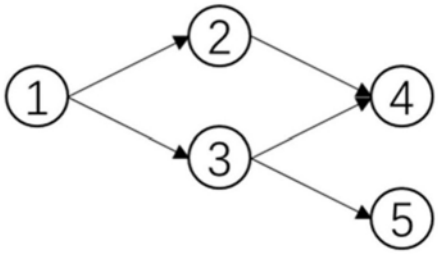
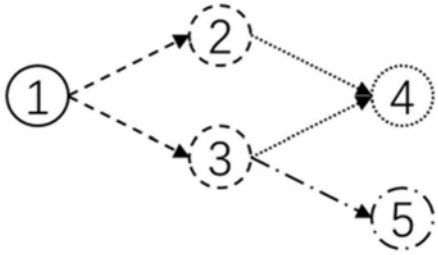


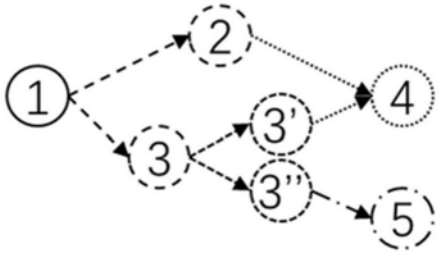
图4



（a）



（b）



（c）

图5

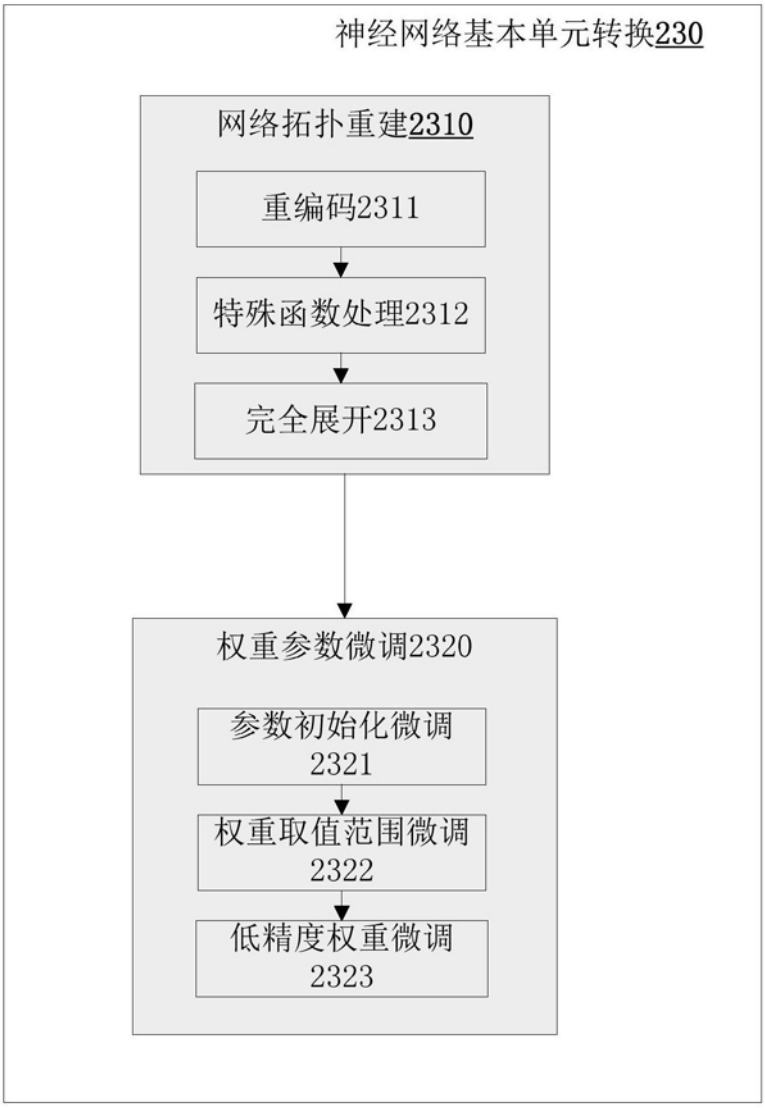

神经网络基本单元转换230
网络拓扑重建2310
重编码2311
特殊函数处理2312
完全展开2313
权重参数微调2320
参数初始化微调2321
权重取值范围微调2322
低精度权重微调2323

图6

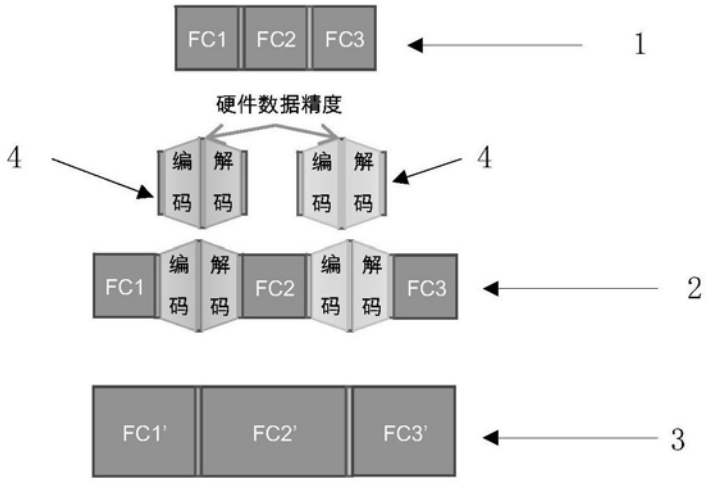

FC1
FC2
FC3
1
硬件数据精度
4
编码
解码
4
FC1
编码
解码
FC2
编码
解码
FC3
2
FC1'
FC2'
FC3'
3

图7

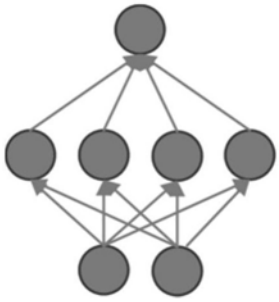


图8

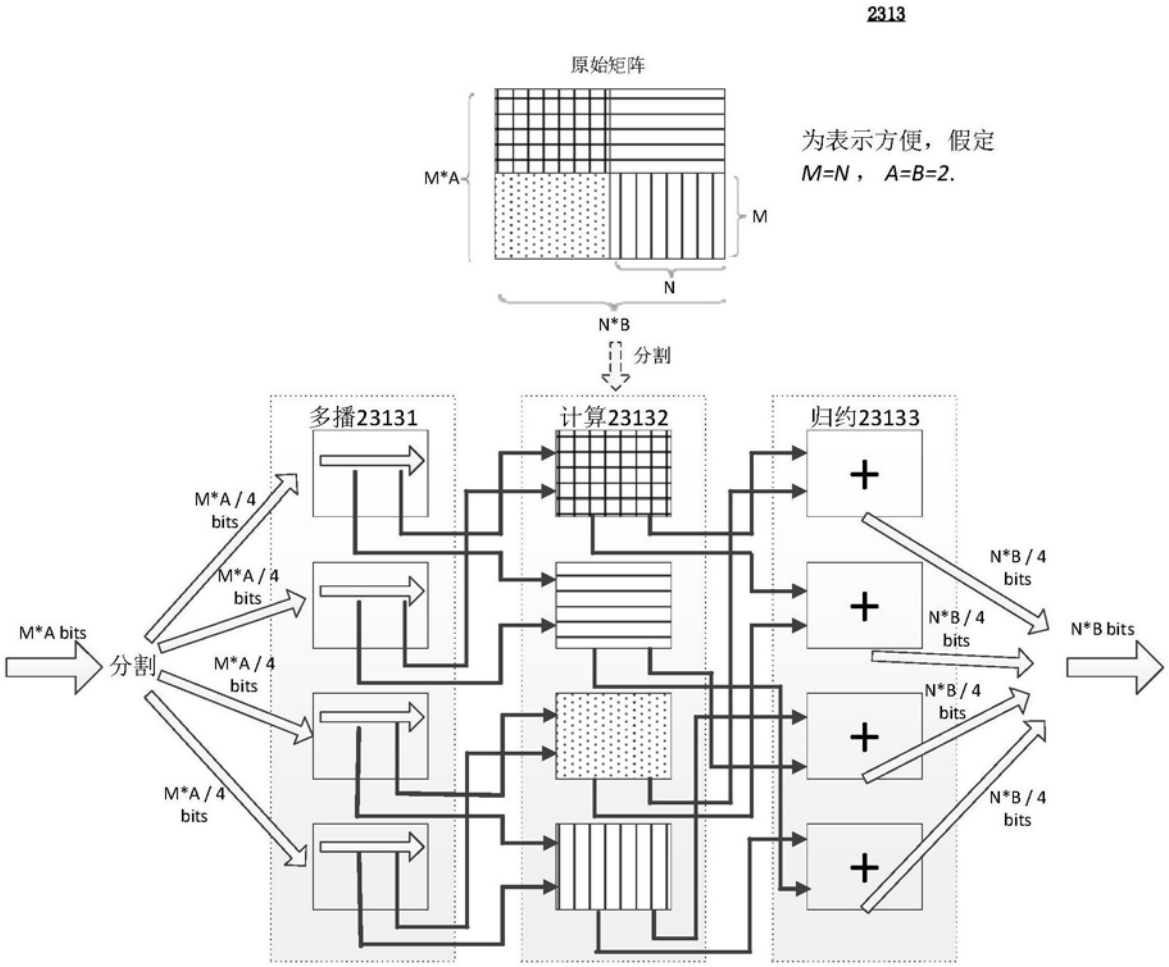

2313
原始矩阵
M*A
M
N
N*B
为表示方便，假定
M=N， A=B=2.
分割
多播23131
计算23132
归约23133
M*A bits
分割
M*A / 4 bits
M*A / 4 bits
M*A / 4 bits
M*A / 4 bits
N*B / 4 bits
N*B / 4 bits
N*B / 4 bits
N*B / 4 bits
N*B bits

图9

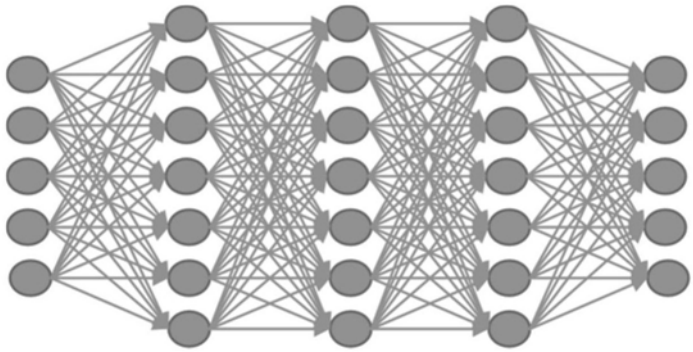


图10

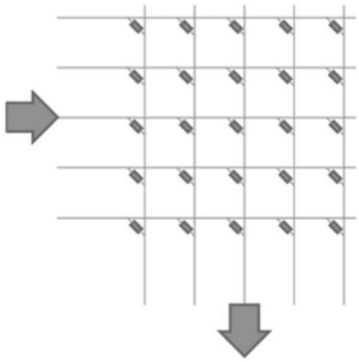


图11