(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 110751265 A
(43)申请公布日 2020.02.04

(21)申请号 201910904649.9

(22)申请日 2019.09.24

(71)申请人 中国科学院深圳先进技术研究院
地址 518055 广东省深圳市南山区深圳大学城学苑大道1068号

(72)发明人 周阳 张涌 宁立 王书强 邬晶晶 姜元爽

(74)专利代理机构 深圳市科进知识产权代理事务所(普通合伙) 44316
代理人 曹卫良

(51)Int.Cl.
G06N 3/04(2006.01)
G06N 3/08(2006.01)
G06F 17/16(2006.01)

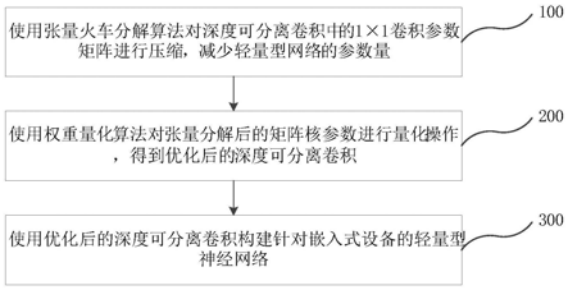
权利要求书2页 说明书9页 附图2页

(54)发明名称

一种轻量型神经网络构建方法、系统及电子设备

(57)摘要

本申请涉及一种轻量型神经网络构建方法、系统及电子设备。包括:步骤a:使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解;步骤b:使用权重量化算法对张量分解后的矩阵核参数进行量化操作,得到优化后的深度可分离卷积;步骤c:使用优化后的深度可分离卷积构建轻量型神经网络。本申请通过使用张量火车分解算法对深度可分离卷积的1×1卷积进行压缩,保持了模型性能的同时,大大减少了深度可分离卷积的参数量。通过使用权重量化算法将张量分解后的核矩阵参数从32bit量化至低比特,减少了模型的计算量并加快了模型的前向推断速度,由此构建的轻量型神经网络能够更好的部署在计算量和存储量有限的嵌入式设备上。



CN 110751265 A

1.一种轻量型神经网络构建方法,其特征在于,包括以下步骤:

步骤a:使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解;

步骤b:使用权重量化算法对张量分解后的矩阵核参数进行量化操作,得到优化后的深度可分离卷积;

步骤c:使用优化后的深度可分离卷积构建轻量型神经网络。

2.根据权利要求1所述的轻量型神经网络构建方法,其特征在于,在所述步骤a中,设卷积核参数矩阵为1×1×M×N,其中M是输入特征图通道数,N是输出特征图通道数,则1×1卷积的总参数量为MN,所述使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解具体为:

步骤a1:提取当前层1×1卷积核参数矩阵,并将其转换为维度为$(m_1n_1,...,m_dn_d)$的张量A,其中$\prod_{i=1}^{d} m_i = M$, $\prod_{i=1}^{d} n_i = N$;

步骤a2:对所述张量A进行张量火车分解,得到核矩阵$G_k[m_k,n_k]$;

步骤a3:将输入特征图的通道数M进行张量火车分解,得到$\mathcal{X}(x,y,m_1,...,m_d)$,其中,$\prod_{i=1}^{d} m_i = M$;

步骤a4:经过张量运算后得到输出特征图$\mathcal{Y}(x,y,n_1,...n_d)$,其中$\prod_{i=1}^{d} n_i = N$,分解后1×1卷积的运算过程表示为:

$$\mathcal{Y}(x,y,n_1,...n_d)=\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{m_1,...,m_d}\mathcal{X}(x,y,m_1,...,m_d)G_0G_1[m_1,n_1]...G_d[m_d,n_d]$$

。

3.根据权利要求2所述的轻量型神经网络构建方法,其特征在于,在所述步骤b中,所述使用权重量化算法对张量分解后的矩阵核参数进行量化操作具体包括:

步骤b1:提取当前层的权值,并计算缩放系数S和零点Z的值;

步骤b2:通过缩放系数S和零点Z计算实际值r对应的量化值q,q=r/S+Z;

步骤b3:输入数据和量化后的权重参数进行对应计算,将得到的结果采用uint32形式保存;

步骤b4:将uint32形式的偏置和步骤b3中的结果相加,并将得到的结果量化为uint8形式;

步骤b5:将步骤b4中所述的uint8形式的结果输入激活函数,得到该层的输出数据,结果为uint8形式。

4.根据权利要求1至3任一项所述的轻量型神经网络构建方法,其特征在于,在所述步骤c中,所述轻量型神经网络的分离卷积核大小均为3×3,激活函数为Relu6。

5.一种轻量型神经网络构建系统,其特征在于,包括:

网络分解模块:用于使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解;

参数量化模块:用于使用权重量化算法对张量分解后的矩阵核参数进行量化操作,得

到优化后的深度可分离卷积；

模型构建模块：用于使用优化后的深度可分离卷积构建轻量型神经网络。

6.根据权利要求5所述的轻量型神经网络构建系统，其特征在于，设卷积核参数矩阵为1×1×M×N，其中M是输入特征图通道数，N是输出特征图通道数，则1×1卷积的总参数量为MN，所述网络分解模块使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解具体为：提取当前层1×1卷积核参数矩阵，并将其转换为维度为(m_1n_1,...,m_dn_d)的张量A，其中$\prod_{i=1}^{d} m_i = M$，$\prod_{i=1}^{d} n_i = N$；对所述张量A进行张量火车分解，得到核矩阵$G_k[m_k, n_k]$；将输入特征图的通道数M进行张量火车分解，得到$\mathcal{X}(x, y, m_1, ..., m_d)$，其中，$\prod_{i=1}^{d} m_i = M$；经过张量运算后得到输出特征图$\mathcal{Y}(x, y, n_1, ...n_d)$，其中$\prod_{i=1}^{d} n_i = N$，分解后1×1卷积的运算过程表示为：

$$\mathcal{Y}(x,y,n_1,...n_d)=\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{m_1,...,m_d}\mathcal{X}(x,y,m_1,...,m_d)G_0G_1[m_1,n_1]...G_d[m_d,n_d]$$

。

7.根据权利要求6所述的轻量型神经网络构建系统，其特征在于，所述参数量化模块使用权重量化算法对张量分解后的矩阵核参数进行量化操作具体包括：1：提取当前层的权值，并计算缩放系数S和零点Z的值；2：通过缩放系数S和零点Z计算实际值r对应的量化值q，q＝r/S+Z；3：输入数据和量化后的权重参数进行对应计算，将得到的结果采用uint32形式保存；4：将uint32形式的偏置和3中的结果相加，并将得到的结果量化为uint8形式；5：将所述uint8形式的结果输入激活函数，得到该层的输出数据，结果为uint8形式。

8.根据权利要求5至7任一项所述的轻量型神经网络构建系统，其特征在于，所述轻量型神经网络的分离卷积核大小均为3×3，激活函数为Relu6。

9.一种电子设备，包括：

至少一个处理器；以及

与所述至少一个处理器通信连接的存储器；其中，

所述存储器存储有可被所述一个处理器执行的指令，所述指令被所述至少一个处理器执行，以使所述至少一个处理器能够执行上述1至4任一项所述的轻量型神经网络构建方法的以下操作：

步骤a：使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解；

步骤b：使用权重量化算法对张量分解后的矩阵核参数进行量化操作，得到优化后的深度可分离卷积；

步骤c：使用优化后的深度可分离卷积构建轻量型神经网络。

一种轻量型神经网络构建方法、系统及电子设备

技术领域

[0001] 本申请属于深度神经网络技术领域，特别涉及一种轻量型神经网络构建方法、系统及电子设备。

背景技术

[0002] 随着深度学习在图像识别、自然语言处理、语音识别等众多领域取得越来越好的效果。为了达到极致的准确率，研究人员普遍会采用更深和更加复杂的网络结构，但这也让神经网络的参数和计算量大大增加，对硬件(处理器，内存，计算卡，带宽)的要求也越来越高，将这些大型的深度神经网络直接部署在计算量和存储量有限的嵌入式设备上并达到可用的速度是很难实现的。随着人工智能在各行各业的应用，将这些大型网络部署在嵌入式设备上的需求越来越大，如何实现神经网络的压缩和加速是人工智能实现产业化必须考虑的一个重要问题。

[0003] 将深度神经网络部署在嵌入式设备上，首先需要考虑这些设备存储空间和计算能力有限的问题，故需要设计非常紧凑和高效的轻量型神经网络结构。MobileNet[Howard A G,Zhu M,Chen B,et al.Mobilenets:Efficient convolutional neural networks for mobile vision applications[J].arXiv preprint arXiv:1704.04861,2017.]是目前最具有代表性的轻量型神经网络，使用了深度可分离卷积(Depth-wise Separable Convolution)代替了传统的卷积运算，在保证模型性能的基础上显著的减少了卷积操作的运算量。深度可分离卷积是将传统卷积操作分为两步：第一步是Depthwise Convolution，一个卷积核只与对应的一个特征图进行卷积；第二步是Pointwise Convolution，卷积核的大小为1×1，即1×1卷积，实现对特征图不同通道之间的线性组合。深度可分离卷积中的1×1卷积可以看做是将一组特征图通过全连接矩阵进行了映射，其中最主要的参数量来自于全连接映射矩阵，其包含了大量的冗余参数(MobileNet中的1×1卷积占据了约75％的参数量和95％的计算量)。

发明内容

[0004] 本申请提供了一种轻量型神经网络构建方法、系统及电子设备，旨在至少在一定程度上解决现有技术中的上述技术问题之一。

[0005] 为了解决上述问题，本申请提供了如下技术方案：

[0006] 一种轻量型神经网络构建方法，包括以下步骤：

[0007] 步骤a：使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解；

[0008] 步骤b：使用权重量化算法对张量分解后的矩阵核参数进行量化操作，得到优化后的深度可分离卷积；

[0009] 步骤c：使用优化后的深度可分离卷积构建轻量型神经网络。

[0010] 本申请实施例采取的技术方案还包括：在所述步骤a中，设卷积核参数矩阵为1×1

×M×N,其中M是输入特征图通道数,N是输出特征图通道数,则1×1卷积的总参数量为MN,所述使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解具体为:

[0011] 步骤a1:提取当前层1×1卷积核参数矩阵,并将其转换为维度为$(m_1n_1,...,m_dn_d)$的张量A,其中$\prod_{i=1}^{d} m_i = M$, $\prod_{i=1}^{d} n_i = N$;

[0012] 步骤a2:对所述张量A进行张量火车分解,得到核矩阵$G_k[m_k,n_k]$;

[0013] 步骤a3:将输入特征图的通道数M进行张量火车分解,得到$\mathcal{X}(x,y,m_1,...,m_d)$,其中,$\prod_{i=1}^{d} m_i = M$;

[0014] 步骤a4:经过张量运算后得到输出特征图$\mathcal{Y}(x,y,n_1,...n_d)$,其中$\prod_{i=1}^{d} n_i = N$,分解后1×1卷积的运算过程表示为:

[0015] $$\mathcal{Y}(x,y,n_1,...n_d)=\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{m_1,...,m_d}\mathcal{X}(x,y,m_1,...,m_d)G_0G_1[m_1,n_1]...G_d[m_d,n_d]$$。

[0016] 本申请实施例采取的技术方案还包括:在所述步骤b中,所述使用权重量化算法对张量分解后的矩阵核参数进行量化操作具体包括:

[0017] 步骤b1:提取当前层的权值,并计算缩放系数S和零点Z的值;

[0018] 步骤b2:通过缩放系数S和零点Z计算实际值r对应的量化值q,q=r/S+Z;

[0019] 步骤b3:输入数据和量化后的权重参数进行对应计算,将得到的结果采用uint32形式保存;

[0020] 步骤b4:将uint32形式的偏置和步骤b3中的结果相加,并将得到的结果量化为uint8形式;

[0021] 步骤b5:将所述uint8形式的结果输入激活函数,得到该层的输出数据,结果为uint8形式。

[0022] 本申请实施例采取的技术方案还包括:在所述步骤c中,所述轻量型神经网络的分离卷积核大小均为3×3,激活函数为Relu6。

[0023] 本申请实施例采取的另一技术方案为:一种轻量型神经网络构建系统,包括:

[0024] 网络分解模块:用于使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解;

[0025] 参数量化模块:用于使用权重量化算法对张量分解后的矩阵核参数进行量化操作,得到优化后的深度可分离卷积;

[0026] 模型构建模块:用于使用优化后的深度可分离卷积构建轻量型神经网络。

[0027] 本申请实施例采取的技术方案还包括:设卷积核参数矩阵为1×1×M×N,其中M是输入特征图通道数,N是输出特征图通道数,则1×1卷积的总参数量为MN,所述网络分解模块使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解具体为:提取当前层1×1卷积核参数矩阵,并将其转换为维度为$(m_1n_1,...,m_dn_d)$的张量A,其中

$\prod_{i=1}^{d} m_i = M$ ，$\prod_{i=1}^{d} n_i = N$ ；对所述张量A进行张量火车分解，得到核矩阵$G_k[m_k,n_k]$；将输入特征图的通道数M进行张量火车分解，得到$\mathcal{X}(x,y,m_1,...,m_d)$，其中，$\prod_{i=1}^{d} m_i = M$；经过张量运算后得到输出特征图$\mathcal{Y}(x,y,n_1,...n_d)$，其中$\prod_{i=1}^{d} n_i = N$，分解后1×1卷积的运算过程表示为：

[0028] $$\mathcal{Y}(x,y,n_1,...n_d)=\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{m_1,...,m_d}\mathcal{X}(x,y,m_1,...,m_d)G_0G_1[m_1,n_1]...G_d[m_d,n_d]$$ 。

[0029] 本申请实施例采取的技术方案还包括：所述参数量化模块使用权重量化算法对张量分解后的矩阵核参数进行量化操作具体包括：1：提取当前层的权值，并计算缩放系数S和零点Z的值；2：通过缩放系数S和零点Z计算实际值r对应的量化值q，q＝r/S+Z；3：输入数据和量化后的权重参数进行对应计算，将得到的结果采用uint32形式保存；4：将uint32形式的偏置和3中的结果相加，并将得到的结果量化为uint8形式；5：将所述uint8形式的结果输入激活函数，得到该层的输出数据，结果为uint8形式。

[0030] 本申请实施例采取的技术方案还包括：所述轻量型神经网络的分离卷积核大小均为3×3，激活函数为Relu6。

[0031] 本申请实施例采取的又一技术方案为：一种电子设备，包括：

[0032] 至少一个处理器；以及

[0033] 与所述至少一个处理器通信连接的存储器；其中，

[0034] 所述存储器存储有可被所述一个处理器执行的指令，所述指令被所述至少一个处理器执行，以使所述至少一个处理器能够执行上述的轻量型神经网络构建方法的以下操作：

[0035] 步骤a：使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解；

[0036] 步骤b：使用权重量化算法对张量分解后的矩阵核参数进行量化操作，得到优化后的深度可分离卷积；

[0037] 步骤c：使用优化后的深度可分离卷积构建轻量型神经网络。

[0038] 相对于现有技术，本申请实施例产生的有益效果在于：本申请实施例的轻量型神经网络构建方法、系统及电子设备通过使用张量火车分解算法对深度可分离卷积的1×1卷积进行压缩，保持了模型性能的同时，大大减少了深度可分离卷积的参数量。通过使用权重量化算法将张量分解后的核矩阵参数从32bit量化至低比特，减少了模型的计算量并加快了模型的前向推断速度。由此构建的轻量型神经网络需要更少的存储空间和算力，能够更好的部署在计算量和存储量有限的嵌入式设备上。

附图说明

[0039] 图1是本申请实施例的轻量型神经网络构建方法的流程图；

[0040] 图2为张量火车分解示意图；

[0041] 图3为引入张量火车分解后的深度可分离卷积示意图；

[0042] 图4为1×1卷积经过量化后运算过程示意图；

[0043] 图5是本申请实施例的轻量型神经网络构建系统的结构示意图；

[0044] 图6是本申请实施例提供的轻量型神经网络构建方法的硬件设备结构示意图。

具体实施方式

[0045] 为了使本申请的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本申请进行进一步详细说明。应当理解，此处所描述的具体实施例仅用以解释本申请，并不用于限定本申请。

[0046] 为了解决现有技术存在的不足，本申请提供一种基于张量火车分解的深度可分离卷积，并结合权重量化算法搭建一套针对嵌入式设备的轻量型神经网络。首先，针对深度可分离卷积中的1×1卷积参数冗余的问题，使用张量火车分解算法对1×1卷积的全连接映射矩阵进行分解，进一步减少轻量型神经网络的参数量；其次，使用权重量化的方法量化张量分解后的矩阵核参数，加快模型的前向推断速度同时减少模型的大小；最后，使用优化后的深度可分离卷积构建针对嵌入式设备的轻量型神经网络。

[0047] 具体的，请参阅图1，是本申请实施例的轻量型神经网络构建方法的流程图。本申请实施例的轻量型神经网络构建方法包括以下步骤：

[0048] 步骤100：使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行压缩，减少轻量型网络的参数量；

[0049] 步骤100中，张量火车分解算法(Tensor-Train)0是一种张量分解算法，可以将高维张量中的每一个元素表示为矩阵连乘(Matrix Product State)的形式，即：

[0050] $A(i_1,i_2,\ldots,i_d)=G_1(i_1)G_2(i_2)\ldots G_d(i_d)$ (1)

[0051] 公式(1)中，$G_k(i_k)$是一个大小为$r_{k-1}\times r_k$的矩阵，r_k为张量火车的分解秩(TT-ranks)，为了保证矩阵连乘的结果是一个标量，$r_0=r_k=1$。

[0052] 图2为张量火车分解示意图。将张量火车分解算法应用在1×1卷积上可以有效的减少深度可分离卷积的参数量，并且保持良好的运算性能。本申请实施例中，使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行压缩的原理为：1×1卷积本质是对输入特征图进行线性组合，实现特征图之间的信息交换，设卷积核参数矩阵为1×1×M×N，其中M是输入特征图通道数，N是输出特征图通道数，那么1×1卷积的总参数量为MN。该参数矩阵是一个全连接矩阵，包含有大量的参数冗余，使用张量火车分解对该参数矩阵进行分解压缩，可以进一步的减少模型的参数量，引入张量火车分解后的深度可分离卷积如图3所示。张量火车分解算法的具体实现步骤包括：

[0053] 步骤101：提取当前层1×1卷积核参数矩阵，并将其转换为维度为$(m_1n_1,\ldots,m_dn_d)$的张量A，其中$\prod_{i=1}^{d} m_i = M$，$\prod_{i=1}^{d} n_i = N$；

[0054] 步骤102：对转换得到的张量A进行张量火车分解，得到的核矩阵为$G_k[m_k,n_k]$；

[0055] 步骤103：将输入特征图的通道数M，以同样的方式分解，得到$\mathcal{X}(x,y,m_1,\ldots,m_d)$，其中，$\prod_{i=1}^{d} m_i = M$；

[0056] 步骤104:经过张量运算后得到输出特征图$\mathcal{Y}(x,y,n_1,...n_d)$,其中$\prod_{i=1}^{d} n_i = N$,分解后1×1卷积的运算过程表示为:

[0057] $$\mathcal{Y}(x,y,n_1,...n_d)=\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{m_1,...,m_d}\mathcal{X}(x,y,m_1,...,m_d)G_0G_1[m_1,n_1]...G_d[m_d,n_d] \quad (2)$$

[0058] 步骤200:使用权重量化算法对张量分解后的矩阵核参数进行量化操作,得到优化后的深度可分离卷积;

[0059] 步骤200中,张量火车分解算法虽然减少了参数量,但是将参数矩阵分解成为了多个矩阵核,增加了矩阵的运算层数,所需要的计算量没有明显减少。因此,本申请将权重量化算法应用在分解后的1×1卷积参数矩阵核上,将32bit参数量化至低比特(本申请实施例中,优选将将32bit参数量化至8bit,具体可根据实际操作进行设定),可以明显的减少计算量,加快模型前向运算速度的同时减少了模型的大小,压缩了模型所需要的存储空间。权重量化算法[Krishnamoorthi R.Quantizing deep convolutional networks for efficient inference:Awhitepaper[J].arXiv preprint arXiv:1806.08342,2018]。是目前使用十分广泛的一种神经网络前向加速技术,将权重参数从32bit量化至低比特,可以显著减少神经网络的运算量,并且能够实现几乎没有精度损失,1×1卷积经过量化后运算过程示意图如图4所示。

[0060] 具体的,权重量化算法的具体步骤包括:

[0061] 步骤201:提取当前层的权值,并计算缩放系数S和零点Z的值;

[0062] 步骤202:通过缩放系数S和零点Z计算实际值r对应的量化值q,q=r/S+Z;

[0063] 步骤203:输入数据和量化后的权重参数进行对应计算,将得到的结果采用uint32形式保存;

[0064] 步骤204:将uint32形式的偏置和步骤203中的结果相加,并将得到的结果使用与步骤201和步骤202同样的方式量化为uint8形式;

[0065] 步骤205:将步骤204中的结果输入激活函数,得到该层的输出数据,结果为uint8形式。

[0066] 步骤300:使用优化后的深度可分离卷积构建针对嵌入式设备的轻量型神经网络;

[0067] 步骤300中,本申请实施例的轻量型神经网络的主要组成为基于张量火车分解的深度可分离卷积,通过将传统的深度可分离卷积替换为基于张量火车分解的深度可分离卷积,分离卷积核大小均为3×3,使用多个小的3×3卷积核的叠加相比于更大的卷积核具有更少的参数和更好的非线性表示性,激活函数使用Relu6,使用了Batch Normalization。并使用权重量化算法将张量火车分解后的深度可分离卷积参数从32bit量化至8bit,维持模型精度的同时,在模型大小和推断速度上相比于MobileNet均有明显的提升。

[0068] 对于ImageNet数据集,本申请实施例的轻量型神经网络的主体架构如下表所示:

[0069] 表1:轻量型神经网络的主体架构

[0070]

层	输入大小	步长	滤波器	通道映射矩阵	通道映射矩阵分解张量	张量火车分解秩
1	224×224×3	2	3×3×3×32			
2	112×112×32	2	3×3×32	32×64	(4×4，2×4，4×4)	(1，8，8，1)
3	56×56×64	1	3×3×64	64×128	(4×8，4×4，4×4)	(1，10，10，1)
4	56×56×128	1	3×3×128	128×128	(4×8，4×4，8×4)	(1，12，12，1)
5	56×56×128	2	3×3×128	128×256	(8×8，4×4，4×8)	(1，16，16，1)
6	28×28×256	1	3×3×256	256×256	(8×8，4×4，8×8)	(1，20，20，1)
7	28×28×256	2	3×3×256	256×512	(8×8，4×8，8×8)	(1，28，28，1)
8-12	14×14×512	1	3×3×512	512×512	(8×8，8×8，8×8)	(1，36，36，1)
13	14×14×512	2	3×3×512	512×1024	(16×8，8×8，8×8)	(1，48，48，1)
14	7×7×1024	1	3×3×1024	1024×1024	(16×8，8×8，8×16)	(1，64，64，1)
15	7×7×1024	1	Avg Pool 7×7			
16	1×1×1024	1	1024×1000			
17	1×1×1000	1	Softmax			

[0071] 压缩前的参数量为3.19M，压缩后的参数量为0.922M，大大减少了深度可分离卷积的参数量。

[0072] 请参阅图5，是本申请实施例的轻量型神经网络构建系统的结构示意图。本申请实施例的轻量型神经网络构建系统包括网络分解模块、参数量化模块和模型构建模块。

[0073] 网络分解模块：用于使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行压缩，减少轻量型网络的参数量；其中，张量火车分解算法是一种张量分解算法，可以将高维张量中的每一个元素表示为矩阵连乘(Matrix Product State)的形式，即：

[0074] $A(i_1,i_2,\ldots,i_d)=G_1(i_1)G_2(i_2)\ldots G_d(i_d)$ (1)

[0075] 公式(1)中，$G_k(i_k)$是一个大小为$r_{k-1}\times r_k$的矩阵，r_k为张量火车的分解秩(TT-ranks)，为了保证矩阵连乘的结果是一个标量，$r_0=r_k=1$。

[0076] 将张量火车分解算法应用在1×1卷积上可以有效的减少深度可分离卷积的参数量，并且保持良好的运算性能。本申请实施例中，使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行压缩的原理为：1×1卷积本质是对输入特征图进行线性组合，实现特征图之间的信息交换，设卷积核参数矩阵为1×1×M×N，其中M是输入特征图通道数，N是输出特征图通道数，那么1×1卷积的总参数量为MN。该参数矩阵是一个全连接矩阵，包含有大量的参数冗余，使用张量火车分解对该参数矩阵进行分解压缩，可以进一步的减少模型的参数量，引入张量火车分解后的深度可分离卷积如图3所示。

[0077] 张量火车分解算法具体包括：提取当前层1×1卷积核参数矩阵，并将其转换为维度为$(m_1n_1,\ldots,m_dn_d)$的张量A，其中$\prod_{i=1}^{d} m_i = M$，$\prod_{i=1}^{d} n_i = N$；对转换得到的张量A进行张量火车分解，得到的核矩阵为$G_k[m_k,n_k]$；将输入特征图的通道数M，以同样的方式分解，得到

$\mathcal{X}(x,y,m_1,...,m_d)$，其中，$\prod_{i=1}^{d} m_i = M$，经过张量运算后得到输出特征图$\mathcal{Y}(x,y,n_1,...n_d)$，其中$\prod_{i=1}^{d} n_i = N$，分解后1×1卷积的运算过程表示为：

[0078] $$\mathcal{Y}(x,y,n_1,...n_d) = \sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{m_1,...,m_d} \mathcal{X}(x,y,m_1,...,m_d) G_0 G_1[m_1,n_1]...G_d[m_d,n_d] \quad (2)$$

[0079] 参数量化模块：用于使用权重量化算法对张量分解后的矩阵核参数进行量化操作，得到优化后的深度可分离卷积；其中，张量火车分解算法虽然减少了参数量，但是将参数矩阵分解成为了多个矩阵核，增加了矩阵的运算层数，所需要的计算量没有明显减少。因此，本申请将权重量化算法应用在分解后的1×1卷积参数矩阵核上，将32bit参数量化至低比特，可以明显的减少计算量，加快模型前向运算速度的同时减少了模型的大小，压缩了模型所需要的存储空间。权重量化算法是目前使用十分广泛的一种神经网络前向加速技术，将权重参数从32bit量化至低比特，可以显著减少神经网络的运算量，并且能够实现几乎没有精度损失。

[0080] 具体的，权重量化算法具体包括：

[0081] 1：提取当前层的权值，并计算缩放系数S和零点Z的值；

[0082] 2：通过缩放系数S和零点Z计算实际值r对应的量化值q，q＝r/S+Z；

[0083] 3：输入数据和量化后的权重参数进行对应计算，将得到的结果采用uint32形式保存；

[0084] 4：将uint32形式的偏置和3中的结果相加，并将得到的结果使用同样的方式量化为uint8形式；

[0085] 5：将4中的结果输入激活函数，得到该层的输出数据，结果为uint8形式。

[0086] 模型构建模块：用于使用优化后的深度可分离卷积构建针对嵌入式设备的轻量型神经网络；本申请实施例的轻量型神经网络的主要组成为基于张量火车分解的深度可分离卷积，通过将传统的深度可分离卷积替换为基于张量火车分解的深度可分离卷积，分离卷积核大小均为3×3，使用多个小的3×3卷积核的叠加相比于更大的卷积核具有更少的参数和更好的非线性表示性，激活函数使用Relu6，使用了Batch Normalization。并使用权重量化算法将张量火车分解后的深度可分离卷积参数从32bit量化至8bit，维持模型精度的同时，在模型大小和推断速度上相比于MobileNet均有明显的提升。

[0087] 对于ImageNet数据集，本申请实施例的轻量型神经网络的主体架构如下表所示：

[0088] 表1：轻量型神经网络的主体架构

[0089]

层	输入大小	步长	滤波器	通道映射矩阵	通道映射矩阵分解张量	张量火车分解秩
1	224×224×3	2	3×3×3×32			
2	112×112×32	2	3×3×32	32×64	(4×4，2×4，4×4)	(1，8，8，1)
3	56×56×64	1	3×3×64	64×128	(4×8，4×4，4×4)	(1，10，10，1)
4	56×56×128	1	3×3×128	128×128	(4×8，4×4，8×4)	(1，12，12，1)
5	56×56×128	2	3×3×128	128×256	(8×8，4×4，4×8)	(1，16，16，1)
6	28×28×256	1	3×3×256	256×256	(8×8，4×4，8×8)	(1，20，20，1)
7	28×28×256	2	3×3×256	256×512	(8×8，4×8，8×8)	(1，28，28，1)
8-12	14×14×512	1	3×3×512	512×512	(8×8，8×8，8×8)	(1，36，36，1)
13	14×14×512	2	3×3×512	512×1024	(16×8，8×8，8×8)	(1，48，48，1)
14	7×7×1024	1	3×3×1024	1024×1024	(16×8，8×8，8×16)	(1，64，64，1)
15	7×7×1024	1	Avg Pool 7×7			
16	1×1×1024	1	1024×1000			
17	1×1×1000	1	Softmax			

[0090] 压缩前的参数量为3.19M，压缩后的参数量为0.922M，大大减少了深度可分离卷积的参数量。

[0091] 图6是本申请实施例提供的轻量型神经网络构建方法的硬件设备结构示意图。如图6所示，该设备包括一个或多个处理器以及存储器。以一个处理器为例，该设备还可以包括：输入系统和输出系统。

[0092] 处理器、存储器、输入系统和输出系统可以通过总线或者其他方式连接，图6中以通过总线连接为例。

[0093] 存储器作为一种非暂态计算机可读存储介质，可用于存储非暂态软件程序、非暂态计算机可执行程序以及模块。处理器通过运行存储在存储器中的非暂态软件程序、指令以及模块，从而执行电子设备的各种功能应用以及数据处理，即实现上述方法实施例的处理方法。

[0094] 存储器可以包括存储程序区和存储数据区，其中，存储程序区可存储操作系统、至少一个功能所需要的应用程序；存储数据区可存储数据等。此外，存储器可以包括高速随机存取存储器，还可以包括非暂态存储器，例如至少一个磁盘存储器件、闪存器件、或其他非暂态固态存储器件。在一些实施例中，存储器可选包括相对于处理器远程设置的存储器，这些远程存储器可以通过网络连接至处理系统。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0095] 输入系统可接收输入的数字或字符信息，以及产生信号输入。输出系统可包括显示屏等显示设备。

[0096] 所述一个或者多个模块存储在所述存储器中，当被所述一个或者多个处理器执行时，执行上述任一方法实施例的以下操作：

[0097] 步骤a：使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解；

[0098] 步骤b:使用权重量化算法对张量分解后的矩阵核参数进行量化操作,得到优化后的深度可分离卷积;

[0099] 步骤c:使用优化后的深度可分离卷积构建轻量型神经网络。

[0100] 上述产品可执行本申请实施例所提供的方法,具备执行方法相应的功能模块和有益效果。未在本实施例中详尽描述的技术细节,可参见本申请实施例提供的方法。

[0101] 本申请实施例提供了一种非暂态(非易失性)计算机存储介质,所述计算机存储介质存储有计算机可执行指令,该计算机可执行指令可执行以下操作:

[0102] 步骤a:使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解;

[0103] 步骤b:使用权重量化算法对张量分解后的矩阵核参数进行量化操作,得到优化后的深度可分离卷积;

[0104] 步骤c:使用优化后的深度可分离卷积构建轻量型神经网络。

[0105] 本申请实施例提供了一种计算机程序产品,所述计算机程序产品包括存储在非暂态计算机可读存储介质上的计算机程序,所述计算机程序包括程序指令,当所述程序指令被计算机执行时,使所述计算机执行以下操作:

[0106] 步骤a:使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行分解;

[0107] 步骤b:使用权重量化算法对张量分解后的矩阵核参数进行量化操作,得到优化后的深度可分离卷积;

[0108] 步骤c:使用优化后的深度可分离卷积构建轻量型神经网络。

[0109] 本申请实施例的轻量型神经网络构建方法、系统及电子设备通过使用张量火车分解算法对深度可分离卷积的1×1卷积进行压缩,保持了模型性能的同时,大大减少了深度可分离卷积的参数量。通过使用权重量化算法将张量分解后的核矩阵参数从32bit量化至低比特,减少了模型的计算量并加快了模型的前向推断速度。由此构建的轻量型神经网络需要更少的存储空间和算力,能够更好的部署在计算量和存储量有限的嵌入式设备上。

[0110] 对所公开的实施例的上述说明,使本领域专业技术人员能够实现或使用本申请。对这些实施例的多种修改对本领域的专业技术人员来说将是显而易见的,本申请中所定义的一般原理可以在不脱离本申请的精神或范围的情况下,在其它实施例中实现。因此,本申请将不会被限制于本申请所示的这些实施例,而是要符合与本申请所公开的原理和新颖特点相一致的最宽的范围。

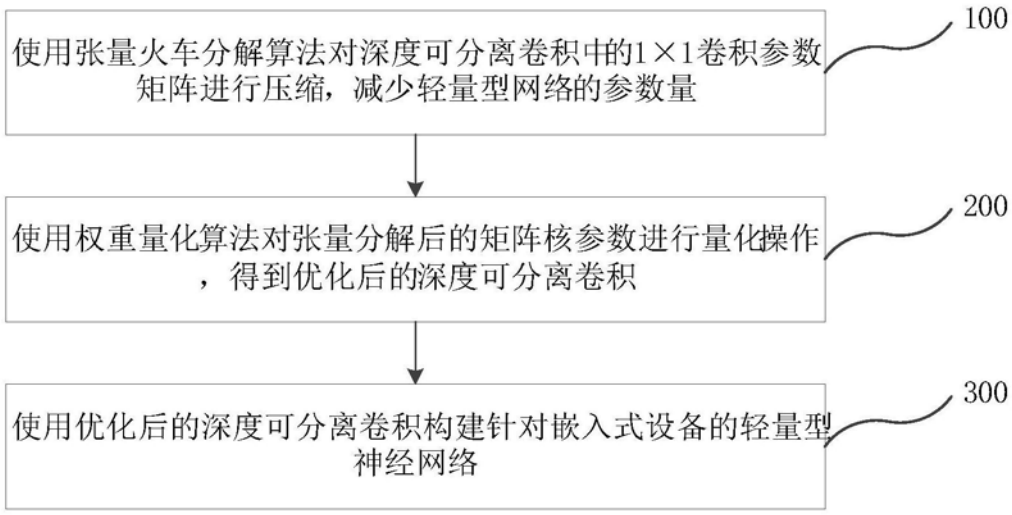

使用张量火车分解算法对深度可分离卷积中的1×1卷积参数矩阵进行压缩，减少轻量型网络的参数量
100
使用权重量化算法对张量分解后的矩阵核参数进行量化操作，得到优化后的深度可分离卷积
200
使用优化后的深度可分离卷积构建针对嵌入式设备的轻量型神经网络
300

图1

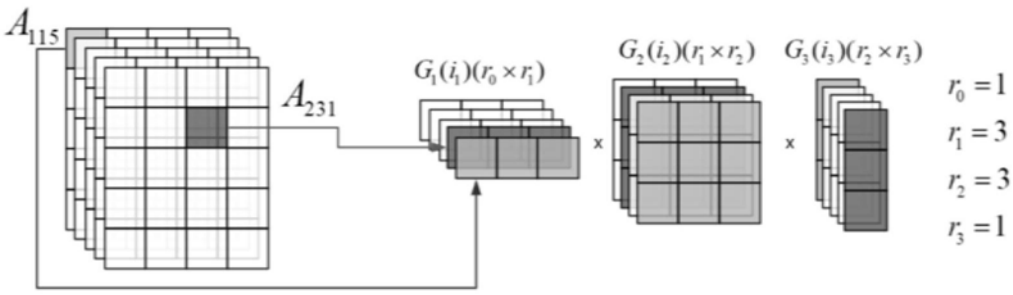

A_{115}
A_{231}
$G_1(i_1)(r_0 \times r_1)$
$G_2(i_2)(r_1 \times r_2)$
$G_3(i_3)(r_2 \times r_3)$
x
x
$r_0 = 1$
$r_1 = 3$
$r_2 = 3$
$r_3 = 1$

图2

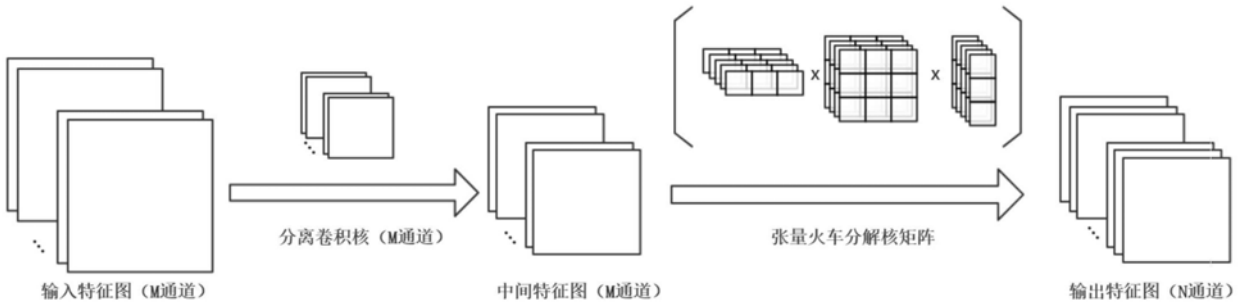

输入特征图（M通道）
分离卷积核（M通道）
中间特征图（M通道）
x
x
张量火车分解核矩阵
输出特征图（N通道）

图3

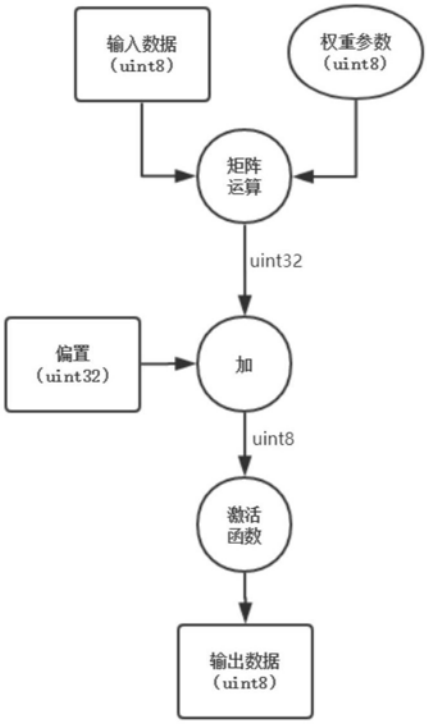

输入数据
(uint8)
权重参数
(uint8)
矩阵
运算
uint32
偏置
(uint32)
加
uint8
激活
函数
输出数据
(uint8)

图4

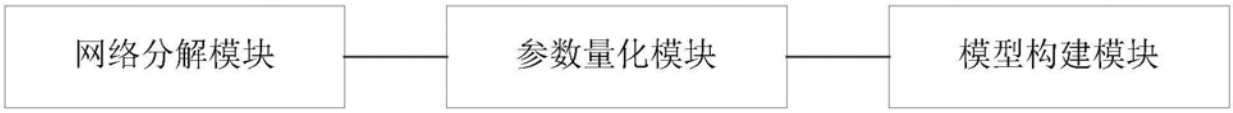

网络分解模块
参数量化模块
模型构建模块

图5

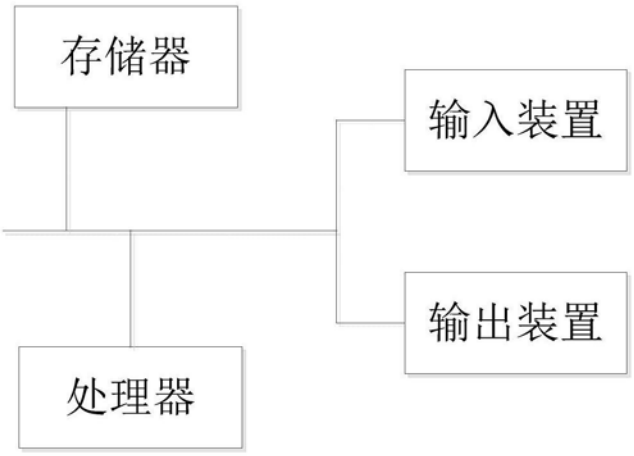

存储器
输入装置
输出装置
处理器

图6