一款人工智能芯片上 FCOS 模型的应用研究

林广栋,黄光红,陆俊峰
(中国电子科技集团公司第三十八研究所,安徽 合肥 230094)

摘　要:人工智能芯片是专门用于高效执行人工智能计算任务的芯片。中国电子科技集团公司第三十八所研制了一款针对边缘侧深度学习模型推理计算的人工智能芯片,主要面向雷达图像目标识别、色选机图像智能处理等应用。该芯片是一个异构的 SOC 芯片,由中央处理核心、神经网络加速核通过片上总线互联形成,峰值算力达到 16TOPS(INT8)。FCOS 模型是一个先进的单阶段无锚框目标检测深度学习模型,该模型首次提出的核心原理已经被一些新的目标检测网络模型采用。该文研究 FCOS 深度学习模型在该人工智能芯片上的部署,并研究片上存储器大小、DDR 带宽、DDR 配置、算力、数据类型等因素对 FCOS 深度学习模型部署的性能和检测效果的影响。可以为深度学习模型部署技术研究人员、人工智能芯片设计人员提供参考。

关键词:人工智能芯片;深度学习;目标检测;DDR;边缘计算
中图分类号:TP32　　**文献标识码**:A　　**文章编号**:1673-629X(2023)05-0009-07
doi:10.3969/j.issn.1673-629X.2023.05.002

Application of FCOS Model on an AI Chip

LIN Guang-dong,HUANG Guang-hong,LU Jun-feng
(The 38th Institute of China Electronics Technology Group Corporation,Hefei 230094,China)

Abstract:AI chip is a kind of chip that can perform artificial intelligence related computing efficiently. The 38th institute of CETC developed an AI chip targeted at deep learning inference tasks at edge devices,and its main application fields includes object detection in radar images,intelligent image processing in color separator devices,etc. The chip is a heterogeneous SOC chip that is composed of central processing unit and neural network accelerator,which are connected together by on-chip bus. Its peak performance achieves 16TOPS(INT8). FCOS is an up-to-date single stage and anchor free object detection deep learning model,the mechanisms firstly proposed by the model have been utilized in some newer object detection models. The application of FCOS on the chip is studied,and the influence of on-chip memory size,DDR bandwidth,DDR configuration,computing power,data type and other factors is thoroughly studied. This work will provide insights to researchers on deployment of deep learning models and also to designers of AI chips.

Key words:AI chip;deep learning;object detection;DDR;edge computing

0　引　言

近年来,深度学习模型在计算机视觉、语音处理等人工智能领域得到了越来越广泛的应用,很多应用领域要求快速且低功耗地完成深度学习模型的推理。例如,在自动驾驶领域,要求深度学习模型在限定的时间内完成图像传感器拍摄的图像中的目标识别任务;在手机等端侧设备中,要求进行图像识别、目标检测的深度学习模型的能耗尽可能小;在大型数据中心,能耗已经成为其成本的重要组成部分,降低深度学习模型在数据中心推理和训练的能耗成为降低数据中心成本的重要因素。然而,深度学习模型的参数量和计算量巨大,在传统的 CPU/DSP 上难以完成高性能且低功耗的推理。因此,专门用于深度学习模型推理的人工智能芯片成为目前研究的热点,且已经有成熟的产品出现[1-2]。中国电子科技集团公司第三十八所研制了一款人工智能芯片,该芯片是一个异构的 SOC(System On Chip)芯片,由支持通用软件的中央处理核心(Central Processing Unit,CPU)和神经网络加速核(Neural Network Accelerator,NNA)构成。其中 CPU 负责一般的软件(如 Linux 或嵌入式操作系统)的运行,而 NNA 负责在 CPU 的调度下完成数值计算密集的神经网络推理任务,两者配合高效地完成深度学习模型的推理。FCOS 模型是目前比较先进的一种单阶段无锚框的目标检测深度学习模型[3],该模型首次提

收稿日期:2022-05-12　　**修回日期**:2022-09-14
基金项目:国家自然科学基金联合基金项目(企业创新发展联合基金)(U19B2041)
作者简介:林广栋(1986-),男,博士,高级工程师,CCF 会员(C0172M),通信作者,研究方向为人工智能芯片基础软件设计。

出了对目标框内的所有特征点输出目标的位置并进行训练的机制。这种机制在后续很多新的目标检测深度学习模型中得到应用。该文研究了 FCOS 模型在该人工智能芯片上的硬件加速技术,介绍了深度学习模型在该人工智能芯片上部署的一般流程,并研究了人工智能芯片的关键配置如算力、DDR 带宽、数据类型对推理性能、最终效果的影响。

1 FCOS 模型介绍

FCOS 是一个一阶段的不基于锚框的目标检测深度学习模型。与其他的一阶段深度学习模型类似,它不需要提取候选区域然后在候选区域上执行图像分类操作。与 yolo 系列基于锚框的目标检测模型相反,它不需要预先设计和定义锚框。它针对输出特征图的每个位置,输出这个位置上的目标的左上角、右下角顶点相对于该位置的偏移。FCOS 模型由骨干(backbone)网络、颈(neck)网络、头(head)网络组成,其中骨干网络负责提取图像不同层次的特征,颈网络负责把不同层次的特征融合,而头网络负责根据不同尺度的特征得到最终的输出。它同样采用了特征金字塔格式的输出,输出层共 5 个分支,分别代表不同尺度下目标的检测信息。FCOS 目标检测模型的骨干网络和颈网络的结构如图 1 所示。

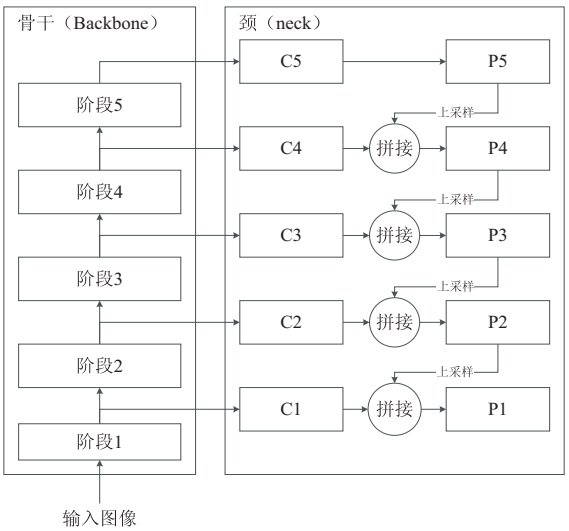

图 1 FCOS 深度学习模型骨干及颈网络示意图

其头部网络的结构如图 2 所示。该图仅仅是输出的五个分支的一个分支示意图,像这样的输出结构在五个分支中都存在。该头部由三类小分支组成,分别是以 softmax 方式处理后代表该位置目标属于各类别的概率的分支,代表该位置处于目标的中心位置的程度的 centerness 分支,代表目标左上角、右下角位置的偏移的分支。若目标有 N 个类别,这三个小分支的通道数分别为:N、1、4。整个模型共 $5*(N+1+4)$ 个输出通道。

FCOS 目标检测深度学习模型提出了一种新的从图像中提取更多训练数据的方法,即位置在真实目标框里的点都可以输出目标的位置,都会进行训练。其于 FCOS 模型的思想,很多新的模型被提出,如 FCOS-3D[4]、TTFNet[5] 等等。相比于 yolo 系列目标检测深度学习模型,FCOS 模型不需要设置锚框,更便于训练,未来将在工业界得到更广泛的应用。

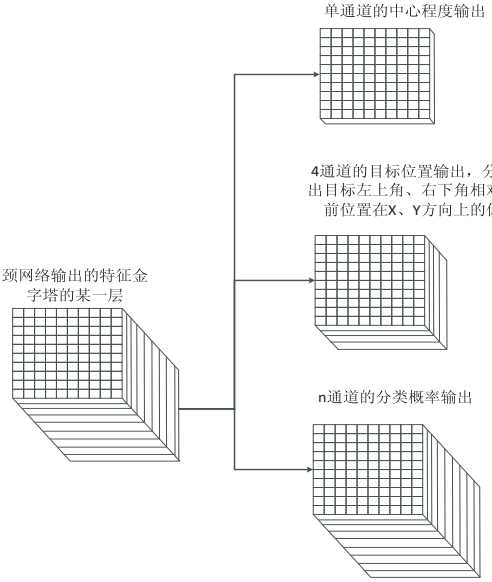

图 2 FCOS 深度学习模型头部网络示意图

2 一款人工智能芯片架构介绍

中国电子科技集团公司第三十八研究所研制了一款人工智能推理芯片,其深度学习推理核心的理论峰值性能达到 16TOPS(INT8),支持 int8、uint8、int16、float16、bfloat16 等数据类型。该芯片的核心 SOC 架构如图 3 所示。

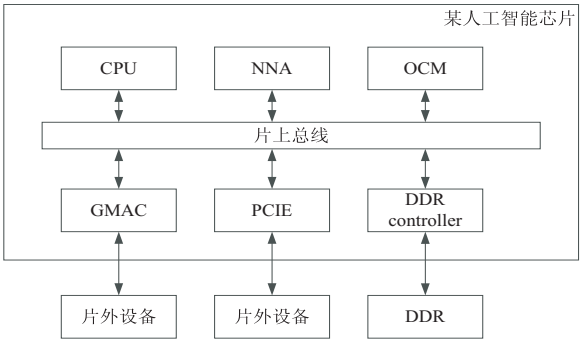

图 3 一款人工智能芯片硬件架构

该芯片是一个由深度学习推理加速核 NNA 与通用处理器 CPU 构成的异构计算系统,两者通过片上总线进行交互。CPU 通过 AHB 总线配置 NNA 的寄存器,而 NNA 通过 AXI 总线访问片上存储器与片外的 DDR,CPU 和 DDR 通过片上存储器与片外的 DDR 共享数据。NNA 内部由 4 个同构的计算核构成,每核理论峰值算力为 4 TOPS(INT8),4 个核可以一起工作完

成同一个任务,也可以分别执行不同的任务。该芯片的高速外设主要包括用于网络通信的以太网接口(Gigabit Media Access Control,GMAC)和用于 PCIE 协议通信的 PCIE 接口。该芯片工作时,首先由 CPU 配置 NNA,使其获取到待执行的神经网络模型的信息,如神经网络模型的结构、权重信息。由 CPU 控制 GMAC 或 PCIE 接口从片外设备(如传感器芯片)获取待处理的输入数据,存储在 DDR 上。之后 CPU 控制 NNA 读取输入数据,执行神经网络模型的推理过程,并把神经网络模型的输出结果写到 DDR 上。之后 CPU 再控制 GMAC 或 PCIE 把计算结果传输到片外设备,进行下一步处理。该芯片配置了一块片上存储器(On Chip Memory,OCM),该存储器相比 DDR 的访问带宽更高。神经网络推理过程中产生的需要反复使用的中间数据,如中间特征图的值,优先存放在 OCM 上,以提高推理效率。

3　FCOS 模型部署步骤

该人工智能芯片提供了完善的软件工具链来支持深度学习模型的部署,包括如下步骤:

①导入:将各种深度学习软件框架生成的模型文件解析为该人工智能芯片内部的模型表示方式,以便后续处理。

②量化:深度学习软件框架中一般用浮点数表示深度学习模型,而如果量化为低位宽的定点数在芯片上进行实时推理,将可提高推理速度[6-7]。对深度学习模型的量化有两种方式:量化敏感的训练[8](Quantization-Aware Training,QAT)、训练后量化[9](Post-Training Quantization,PTQ)。前者在量化完成之后再使用训练数据对量化后的模型进行精调。后者在训练完成之后根据一些测量数据对激活度的范围进行测量后直接量化。该人工智能芯片配套工具链使用的是训练后量化方法。量化时,需要提供少量测试数据,软件工具链会对这些输入数据执行推理过程,以得到深度学习模型各层特征图的取值范围,再进行量化,以使量化后的定点数最大程度地覆盖原始模型的浮点数的取值范围。深度学习模型的量化分为不同的层次,包括逐层量化[10]、分组量化[11]、逐通道量化[12]等等。该人工智能芯片的软件工具链的量化算法均使用逐层量化的方式。深度学习模型在芯片中的量化推理方式按量化参数是否动态变化又可分为两类:动态量化[13]、静态量化[14]。前者的量化参数会在运行时根据实际激活度的变化范围进行调整;而后者的量化参数在推理前确定,并在运行时保持不变。该人工智能芯片的软件工具链的量化方式是静态量化方式。

③优化:该人工智能芯片的软件工具链内部以计算图的方式表示深度学习模型,基于计算图,可以执行如算子合并、冗余计算删除等计算图优化操作,在不降低精度的基础上提高性能。

④导出:把经过量化、优化后的深度学习模型保存下来,输出为模型文件。该人工智能芯片的模型文件同时包含模型的结构与量化后的权重。

⑤推理:芯片上的驱动在应用程序的调用下,加载并解析模型文件,根据具体的硬件配置对计算图执行进一步的优化,并执行实时的模型推理任务。

4　量化方式

由于深度学习模型中的权重存在一定冗余性,因此把权重及激活度量化为低位宽的数据,可以在不明显降低模型精度的前提下减少模型的计算量、减小模型的大小,进而减少模型推理时对于片外数据传输带宽的需求,最终提高模型推理的效率。量化算法主要分为两类:对称量化[15]和非对称量化[16]。当以对称量化算法量化为 8 位时,数据类型称为 int8;当以非对称量化算法量化为 8 位时,数据类型称为 uint8;当以对称量化算法量化为 16 位时,数据类型称为 int16。

4.1　INT8

量化为 INT8 方式时,由浮点数转换为定点数的计算方式为:

首先计算中间值:

$$data = round(fdata * 2^{fl})$$

然后计算最终量化值:

$$qdata = \begin{cases} data & if \quad -128 \leqslant data \leqslant 127 \\ 127 & if \quad 127 < data \\ -128 & if \quad data < -128 \end{cases}$$

而由量化后的 INT8 值计算原始浮点值的方式如下:

$$fdata = qdata * 2^{-fl}$$

其中,fl 是进行 INT8 量化后的常数,每一层的权重与每一层的激活度在量化后有不同的 fl 值,分别根据该层权重与激活度的分布计算得到。其中权重的 fl 的计算方法如下:

$$fl = 7 - \lceil \log_2(\max(\mathrm{abs}(w))) \rceil$$

其中,$\max(\mathrm{abs}(w))$ 代表一层的权重的绝对值的最大值。激活度的 fl 按类似的方法根据激活度的分布计算得到。

4.2　UINT8

UINT8 型量化将权重和激活度都量化为无符号的 8 位数,量化后的数值范围在 0 ~ 255 之间。由浮点数计算 UINT8 量化数的计算方式如下:

首先计算中间值:

$$data = round(fdata/scale + zeropoint)$$

然后计算最终量化值：

$$qdata=\begin{cases}255 & \text{if} \quad 255<data\\ data & \text{if} \quad 0\leqslant data\leqslant 255\\ 0 & \text{if} \quad data<0\end{cases}$$

而由量化后的 UINT8 型数据转换为浮点数据的计算方式为：

$$fdata=(qdata-zeropoint)*scale$$

其中，scale 和 zeropoint 为根据特定算法计算得到的缩放因子与零点。以计算某一层的权重的 scale 和 zeropoint 为例，记神经网络某层的权重的最大值为 max(w)，最小值为 min(w)，则该层的权重量化为 UINT8 时的 scale 和 zeropoint 的计算方式如下：

$$scale=\begin{cases}\max(w)/255 & \text{if} \quad 0<\min(w)\\ \min(\text{abs}(w))/255 & \text{if} \quad \max(w)<0\\ (\max(w)-\min(w))/255 & \text{if} \quad \text{其他}\end{cases}$$

$$zeropoint=\begin{cases}0 & \text{if} \quad 0<\min(w)\\ 255 & \text{if} \quad \max(w)<0\\ -\min(w)/scale & \text{if} \quad \text{其他}\end{cases}$$

4.3 INT16

量化为 INT16 方式时，由浮点数转换为定点数的计算方式为：

首先计算中间值：

$$data=\text{round}(fdata*2^{fl})$$

然后计算最终量化值：

$$qdata=\begin{cases}data & \text{if} \quad -32\,768\leqslant data\leqslant 32\,767\\ 32\,767 & \text{if} \quad 32\,767<data\\ -32\,768 & \text{if} \quad data<-32\,768\end{cases}$$

而由量化后的 INT8 值计算原始浮点值的方式如下：

$$fdata=qdata*2^{-fl}$$

其中，fl 是进行 INT8 量化后的常数，每一层的权重与每一层的激活度在量化后有不同的 fl 值，分别根据该层权重与激活度的分布计算得到。其中权重的 fl 的计算方法如下：

$$fl=15-\lceil\log_2(\max(\text{abs}(w)))\rceil$$

其中，max(abs(w)) 代表一层的权重的绝对值的最大值。激活度的 fl 按类似的方法根据激活度的分布计算得到。

4.4 FLOAT16

FLOAT16 是 IEEE 规定的标准数据格式，共 16 位，各位的含义如表 1 所示。

表 1 FLOAT16 数据类型

符号	指数	尾数
1 位	5 位	10 位

4.5 BFLOAT16

在深度学习领域，由于网络模型中存在大量参数，这些参数具有大量的冗余性，精确地表示这些参数的重要性降低。在深度学习模型推理领域，人们开始使用 bfloat16 数据类型，这种数据类型相对于常规的 float16 数据类型降低了尾数的位宽，增加了指数的位宽，其效果是增加了其表示的数值的范围，减少了表示的精度。这种数据类型能在与 float16 相同的数据位宽下以较低的精度表示更大的数据范围，比较适合深度学习领域。这种数据类型各位的含义如表 2 所示。

表 2 BFLOAT16 数据类型

符号	指数	尾数
1 位	8 位	7 位

5 实 验

以下实验基于中国电子科技集团公司开发的针对该人工智能芯片的演示板卡完成，该演示板卡实物图如图 4 所示。

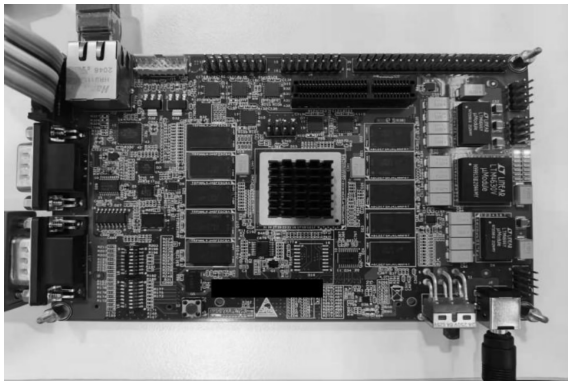

图 4 一款人工智能芯片演示板卡实物图

以下实验中，FCOS 模型的输入图像宽度为1 216，高度为 800，总卷积计算量约为 138 GOPS。

5.1 片上存储器的影响

该人工智能芯片内部设置了 4 MB 大小的片上存储器。片上存储器的访问延迟比片外的 DDR 小得多，并且其带宽可以达到片上总线传输带宽的上限。将 FOCS 模型量化为精度比较高的 INT16 数据类型，分别控制使用不同大小的片上存储器，对性能的影响如表 3 所示。

表 3 片上存储器大小对 FCOS 模型推理时间的影响

片上存储器大小/MB	推理时间/ms
0	1 192
1	1 181
2	1 174
3	1 163
4	1 148

由表 3 可以看出,使用芯片内部的片上存储器可以提高深度学习模型的推理速度。当然,片上存储器会增加芯片的面积与功耗,其容量不可能设置太大,需要在推理性能与芯片的面积和功耗之间进行平衡。

5.2　DDR 带宽的影响

深度学习模型推理时,其中间层的通道数量很大,使得中间层特征图无法在片上存储器全部存储,需要在片外容量更大的 DDR 中暂存。这就导致深度学习模型推理时需要进行大量片上数据与片外数据的传输,因此,DDR 的带宽对模型推理的性能影响很大。将 FCOS 模型量化为 INT16 数据类型,然后分别配置 DDR 控制器的频率为不同的数值,在不同的 DDR 带宽下进行推理,模型推理的性能如表 4 所示。该芯片使用 DDR 控制器数据位宽为 64 位,理论峰值带宽(bandwidth)与频率(frequency)的关系为:

$$bandwidth = frequency * 64$$

其中,频率的单位为 MHz,而带宽的单位为 Mbit/s。

表 4　DDR 带宽对 FCOS 模型推理时间的影响

DDR 频率/MHz	推理时间/ms
3 200	915
2 640	955
2 400	992
2 000	1 081
1 600	1 224

可见,随着 DDR 频率的降低,推理性能也呈现明显的降低。显然,DDR 带宽对推理性能有着重要的影响。

5.3　DDR 配置的影响

DDR 有很多配置选项,包括配置各 AXI 端口的优先级、带宽限制、是否使能 bank group、写命令重排、命令队列选择等等。对 DDR 控制器的不同属性进行配置的寄存器数量多达三百多个。同 DDR 的带宽配置一样,DDR 的各项配置也会对推理性能产生影响。该文无法穷尽所有的 DDR 配置,仅就是否使能 bank group、是否打开写重排功能、是否打开命令选择功能三个选项进行实验,检验这些配置对推理性能的影响。表 5 为几种典型的 DDR 配置及不同的 DDR 带宽下 FCOS 模型的推理性能,该表中的数据均是在模型量化为 INT16 数据类型、使用 4 核推理、片上存储器容量为 4 MB 时统计出来的。

其中 bank group 是 DDR4 设备专用的概念,它把区分 bank group 的第[0]地址放到区分 DRR 颗粒“列”的地址位中,使 DDR 控制器同时维护两个 bank group 的状态,可以以更高的效率支持连续的 burst 读写。在本芯片的 DDR 配置中,支持 bank group 时,bank group 的第[0]位位于软件视角的地址的第[6]位。当 DDR 收到的 burst 请求大于 64 byte 时,使能 bank group 的效果更好。但实验表明,使能 bank group 这个功能(bg_rotate_en)反而会降低性能,这是因为 NNA 发出的 burst 请求大小一般为 64 个 byte 或更小,而很少发出更大 burst 请求,这是由 NNA 的核心架构决定的。因此,NNA 无法利用 bank group 的优势。

表 5　不同 DDR 配置下 FCOS 模型推理时间　ms

DDR 频率/MHz	使能 bank group	×	使能 bank group	√	使能 bank group	×	使能 bank group	×
	打开写重排	√	打开写重排	√	打开写重排	×	打开写重排	√
DDR 配置	打开命令队列选择	√	打开命令队列选择	√	打开命令队列选择	√	打开命令队列选择	×
3 200	916		944		965		982	
2 640	955		986		1 014		1 034	
2 400	992		1 026		1 058		1 083	
2 000	1 081		1 124		1 168		1 202	
1 600	1 224		1 277		1 336		1 388	

DDR 控制器维护了一个命令队列,按照一定的逻辑把来自不同总线端口的访问请求放入队列中,并支持按一定的逻辑从队列头部的 4 个命令中选择最适当的命令发送给 DDR 颗粒。DDR 控制器一般根据 bank 是否冲突、地址是否冲突等规则决定从命令队列前 4 个命令中取出命令的顺序。若关闭命令队列选择(in_order_accept)功能,则 DDR 控制器总是选择队列头部的命令发送给颗粒。实验表明,打开命令队列选择的推理性能更好,这是因为 DDR 控制器会在队列头部的命令因为颗粒未准备好等原因而无法执行时,选择队列头部前 4 个命令中的其他命令执行,从而提高了效率。

DDR 控制器对来自不同端口的写命令有三种策略(wr_order_req):(1)不论是否是来自相同的总线端口的写请求,也不论命令 ID 是否相同,都可以改变写的顺序;(2)来自相同总线端口的带有相同写命令 ID 的请求不会被重排,其他的写命令可以被重排;(3)只要是来自相同总线端口的写请求,都会按发送到 DDR 控制器的顺序执行,不会被重排;来自不同总线端口的写请求会被重排。显然,根据 DDR 颗粒的状态及时改变写命令的执行顺序,将可以提高写命令的执行效率。实验表明,写重排功能打开时的推理性能要优于写重排关闭时的推理性能。

5.4　算力的影响

该人工智能芯片中,深度学习推理加速核内部由

4 个结构相同的核构成,每个核的理论峰值算力为 4 TOPS(int8),这 4 个核可以组合配置为不同的算力。作为深度学习推理的核心部件,算力的配置显然也对推理性能产生影响。表 6 统计出不同算力配置下 FCOS 模型的推理性能,此表中的数据均是模型量化为 INT16 数据类型、片上存储器容量设为 4 MB 时统计出的。

表 6 算力配置对 FCOS 模型推理时间的影响

算力配置	推理时间/ms
单核	3 196
双核	1 852
四核	1 142

显然,算力配置越高,推理性能越强。但推理性能与算力之间并不是线性关系。例如,四核配置下的推理时间并不是单核配置下推理时间的 1/4,主要原因有两点:(1)算力提高,计算需要的数据量线性增长,对带宽的要求也相应提高,但芯片的整体带宽不变,因此推理性能不能线性增长;(2)核数量增加,需要额外的操作进行特征图的切分与计算结果的合并、核之间计算的同步,带来额外的负担。因此,推理性能并不能随着算力配置的增加而线性增加。

5.5 数据类型的影响

部分硬件电路可复用为支持不同的数据类型。例如,一个计算 INT16 乘法的电路可以复用为 4 个 INT8 乘法的电路。同样,该电路也可复用于计算浮点数据类型尾数的乘法。理论上,本芯片计算 INT8 数据类型的算力是计算 INT16 数据类型时算力的 4 倍。由于芯片带宽、片上存储器容量等其他因素限制,实际执行推理运算时,INT8 与 INT16 数据类型的表现并不完全是 4 倍的关系。在 CPU 运行在 1 200 MHz、NNA 运行在 660 MHz、DDR 运行在 2 400 MHz 频率下,片上存储器容量固定为 4 MB 时,量化为各种数据类型的 FCOS 模型的运行速度如表 7 所示。

表 7 数据类型对 FCOS 模型推理时间的影响

数据类型	时间/ms
INT8	389
UINT8	381
INT16	1 142
FLOAT16	1 134
BFLOAT16	1 209

从推理性能上看,int8 和 uint8 数据类型的推理性能几乎相同,int16 和 float16 的推理性能约是 int8 和 uint8 的 3 ~4 倍。Bfloat16 由于需要在推理前和推理后执行向常规数据类型的转换,性能最差。

使用不同数据类型对 FCOS 模型进行量化后进行目标检测的实际效果如表 8 所示。

表 8 FCOS 模型量化为不同数据类型后的实际目标检测效果

数据类型	检测效果
INT8	
UINT8	
INT16	
FLOAT16	
BFLOAT1	

从以上结果可以看出,量化为 BFLOAT 和 INT16 数量类型的 FCOS 模型可以检测出最左侧一列中从网球到网球拍、人三种不同尺度的物体,表现最好。最终推理的结果精度上看,效果从好到差依次为:bfloat16

=int16>float16>uint8>int8。其中 int16 数据类型等效于对同一层的特征图,使用统一的指数来表示,由于其尾数位数大于 float16 数据类型的尾数位数(10 位),因此,其最终的效果比 float16 更高。

6　结束语

该文介绍了 FCOS 目标检测模型的基本网络结构。同时介绍了一款人工智能芯片的基本硬件结构。研究了把 FCOS 深度学习模型应用到该人工智能芯片的方法,并研究了片上存储器大小、DDR 带宽、DDR 配置、算力、不同的量化算法等因素对推理效果的影响。研究表明,从对推理精度的影响来看,int16 量化方法和 bfloat16 数据类型的精度最高,float16、uint8 数据类型的精度依次降低,int8 最差。从对推理时间的影响来看,bfloat16 数据类型的效果最差,int16 和 float16 的性能次之,int8 和 uint8 的推理时间最短。研究结果证实,片上存储器容量越大、DDR 带宽都对推理时间产生重要的影响,片上存储器容量越大、DDR 带宽越大,推理时间越短;反之则越长。另外,研究还表明,DDR 的配置,如是否使能 bank group、是否使能命令队列选择功能、是否支持写重排,也会对推理时间产生影响,但影响的程度不如 DDR 带宽的影响。研究成果将为人工智能芯片、深度学习模型推理算法的研究者提供参考。

参考文献:

[1] CHEN Yiran, XIE Yuan, SONG Linghao, et al. A survey of accelerator architectures for deep neural networks[J]. Engineering, 2020, 6(3): 264-274.

[2] REUTHER A, MICHALEAS P, JONES M, et al. AI accelerator survey and trends[J]. arXiv: 2109.08957, 2021.

[3] TIAN Z, SHEN C, CHEN H, et al. FCOS: fully convolutional one-stage object detection[C]//2019 IEEE/CVF international conference on computer vision (ICCV). Seoul: IEEE, 2019: 9626-9635.

[4] WANG T, ZHU X, PANG J, et al. FCOS3D: fully convolutional one-stage monocular 3D object detection[J]. arXiv: 2104.10956, 2021.

[5] LIU Z, ZHENG T, XU G, et al. Training-time-friendly network for real-time object detection[J]. arXiv: 1909.00700, 2019.

[6] GHOLAMI A, KIM S, DONG Z, et al. A survey of quantization methods for efficient neural network inference[J]. arXiv: 2103.13630, 2021.

[7] JACOB B, KLIGYS S, CHEN B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference[J]. arXiv: 1712.05877, 2017.

[8] HAWKS B, DUARTE J, FRASER N J, et al. Ps and qs: quantization-aware pruning for efficient low latency neural network inference[J]. arXiv: 2102.11289, 2021.

[9] CAI Yaohui, YAO Zhewei, DONG Zhen, et al. Zeroq: a novel zero shot quantization framework[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. Seattle: IEEE, 2020: 13169-13178.

[10] KRISHNAMOORTHI R. Quantizing deep convolutional networks for efficient inference: a whitepaper[J]. arXiv: 1806.08342, 2018.

[11] SHEN Sheng, DONG Zhen, YE Jiayu, et al. Q-BERT: hessian based ultra low precision quantization of bert[C]//Proceedings of the AAAI conference on artificial intelligence. New York: AAAI, 2020: 8815-8821.

[12] JACOB B, KLIGYS S, CHEN B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). Salt Lake City: IEEE, 2018: 2704-2713.

[13] YAO Zhewei, DONG Zhen, ZHENG Zhangcheng, et al. Hawqv3: dyadic neural network quantization[J]. arXiv: 2011.10680, 2020.

[14] CHOUKROUN Y, KRAVCHIK E, YANG F, et al. Low-bit quantization of neural networks for efficient inference[C]//2019 IEEE/CVF international conference on computer vision workshop (ICCVW). Seoul: IEEE, 2019: 3009-3018.

[15] WU Hao, JUDD P, ZHANG Xiaojie, et al. Integer quantization for deep learning inference: principles and empirical evaluation[J]. arXiv: 2004.09602, 2020.

[16] BHALGAT Y, LEE J, NAGEL M, et al. Lsq+: improving low-bit quantization through learnable offsets and better initialization[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. Seattle: IEEE, 2020: 696-697.