基于 FPGA 的卷积神经网络加速技术研究

薛慧敏[1] 李坤坤[2] 眭畅豪[1]
XUE Huimin LI Kunkun SUI Changhao

摘 要 实现高性能且低功耗的神经网络功能具有重要的意义。为了让神经网络加速预测并使其高效节能，设计了卷积神经网络加速器。首先采集信息对训练好的 CNN 神经网络模型使用剪枝和量化对网络模型共同作用进行压缩；然后通过研究数据的存储方式、流动过程、CNN 网络的层结构等多个模块分别提出优化方案；最后在 Xilinx 的 UltraSacle+ 系列的 ZCU102 板子上设计 CNN 网络加速器，使得所设计的加速器性能在前人的基础上有所改进。实验结果显示，CNN 加速器的 FPGA 比 CPU 的计算速度提高了 314.55 倍，比 GPU 的能量效率提高了 1.39 倍，为卷积网络模型加速器以及门控单元 GRU 网络等其他网络模型的加速提供了有效参考。

关键词 卷积神经网络；FPGA；硬件加速；模型压缩

doi：10.3969/j.issn.1672-9528.2024.04.044

0 引言

随着计算机算力的快速提升和处理现实问题的需要，深度学习（deep learning）变得越来越火，而神经网络（neural networks）作为深度学习主要实现方式之一，在人工智能领域[1]取得了历史性的进展，它在从物体检测[2-6]到自动驾驶[7-10]的各个方面都表现出巨大的潜力。深度神经网络的迅速发展带来了很高的预测准确率，但是在具体的工程应用中却面临着诸多问题。其中最主要的是大量的训练数据和推断测试数据导致了模型训练和推断过程中很高的计算复杂度，进而导致计算速度受限和大量的内存占用而带来的高功耗问题，这成为了神经网络在部署过程中的一大瓶颈。

针对以上问题，本文设计了一种卷积神经网络加速器，在确定神经网络的精度不变或者改变较小的条件下优化算法，再以硬件友好的方式表示压缩模型，大幅度地减少了对存储资源需求和计算空间的占用，达到对神经网络加速的目的。

1 加速策略

1.1 剪枝

本文使用多轮迭代剪枝的措施，最开始剪枝的时候不会将剪枝率设得太大，从剪枝率 S_i 开始，S_i 一般设为 0，逐步提高剪枝率，直到达到预设剪枝率 S_f，这样的话不会将重要的权重值在最开始就被剪掉，而造成无法挽回的精度和准确率的损失。剪枝率的计算公式为：

$$S_t = S_f + (S_i - S_f)\left(1 - \frac{t - t_0}{n\Delta t}\right)^3 \quad (1)$$

$$t \in \{t_0, t_0 + \Delta t, ..., t_0 + n\Delta t\}$$

1.2 定点量化

用高位宽的浮点数表示神经网络的权重参数，需要巨大的存储资源，对剪枝后的神经网络按照要求减小每个权重的位数以减小其所需要的存储资源，这个操作称为量化。使用低位宽的定点数表示。在 FPGA 平台上，对定点数进行运算消耗的时间和空间要少于浮点数运算，因此，在整体相等的空间上，定点数的运算容易获得更强的并行性，可以大幅度地提高计算的速度；同时，也可以有效地减少计算的能量消耗，减小了 FPGA 平台的能量消耗。

1.3 量化公式的设计

本文使用线性量化。首先对权值进行适当的缩放，让权重值分布在 [0,1] 之间，缩放公式如式（3）所示，其中缩放因子 scale 通过式（2）得出。

$$scale = np.max\,(np.abs(x)) \quad (2)$$

$$zoom = x/scale \quad (3)$$

根据需要量化的位数，对缩放之后的权重进行量化，量化公式为：

$$Q = np.int8(np.around(zoom \times 8)) \quad (4)$$

再将量化后的值进行反缩放，使权重恢复到缩放前的原始值，其公式为：

1. 中国航空工业集团西安航空计算技术研究所 陕西西安 710000
2. 中国人民解放军 63660 部队 河南洛阳 471099

$$Q' = np.int8(np.around(zoom \times 8))/8 \times MAX \quad (5)$$

以上的量化过程一共包括了缩放 - 量化 - 反缩放三个步骤。本文使用该量化过程将 32 位浮点数表示的权重参数变为使用 8 位定点数表示，缩小了存储权重所需要的空间，减少了计算所需要的时间，对神经网络模型的压缩起到了重要的作用。

1.4 BN 层的归一化

BN 层通过线性计算使得数值的分布在绝对值较小的地方概率比较大，而在绝对值较大的地方概率变小，减小训练阶段的梯度消失现象。当数据通过卷积运算之后，卷积层的输出进入 BN 层，将一些相邻的模块进行归一化，然后再进入激活函数模块。BN 归一化也被称为 BN 融合，该操作减小了网络的数据量，提高了运算的速度。BN 归一化的计算公式为：

$$y_{bn} = \gamma\left(\frac{x_i-\mu}{\sqrt{\sigma^2+\epsilon}}\right) + \beta \quad (6)$$

式中：x_i 是输入数据的特征值，μ 是这批数据的均值，σ 是其方差，ϵ是一个特别小的数，主要是为了避免分数中分母为零，γ 和 β 是常数，起线性调整作用。对公式（6）进行分解得到：

$$y_{bn} = \gamma\left(\frac{x_i}{\sqrt{\sigma^2+\epsilon}}\right) - \gamma\left(\frac{\mu}{\sqrt{\sigma^2+\epsilon}}\right) + \beta \quad (7)$$

然后对公式（7）进行移项，得到公式（8）。

$$y_{bn} = \gamma\left(\frac{x_i}{\sqrt{\sigma^2+\epsilon}}\right) + (\beta - \gamma\left(\frac{\mu}{\sqrt{\sigma^2+\epsilon}}\right)) \quad (8)$$

将公式（8）中的分子和乘数进行位置变换得到公式（9）。

$$y_{bn} = \left(\frac{\gamma}{\sqrt{\sigma^2+\epsilon}}\right)x_i + (\beta - \gamma\left(\frac{\mu}{\sqrt{\sigma^2+\epsilon}}\right)) \quad (9)$$

使用简单的符号代表公式（9）中的参数，得到的 BN 归一化的最终简化公式如公式（10），这样就可以很容易地看出 BN 层的归一化其实是线性的计算过程。

$$y_{bn} = ax_i + b \quad (10)$$

式中：$a = \frac{\gamma}{\sqrt{\sigma^2+\epsilon}}$，$b = (\beta - \gamma\left(\frac{\mu}{\sqrt{\sigma^2+\epsilon}}\right))$，$a$ 和 b 在训练期间得出，推理阶段是常数，通用的加速器适用于任何尺寸的 BN 运算。

由此可以发现，BN 操作是一个线性操作，和卷积操作的性质一样，所以可以将 BN 操作和卷积层的卷积操作进行合并，给卷积操作的缩放因子乘以 BN 操作的缩放因子，将卷积层的偏置乘以 BN 层的缩放因子再和 BN 层的偏置加起来即可。这样的话就完成了 BN 层和卷积层的归一化，达到了减少计算量的目的。

1.5 量化位数的选择

本文将 32 位的浮点数表示的剪枝后的神经网络参数逐步减小位数，使用 8 位定点数表示和 32 位浮点数表示的预测准确率相比没有太大的降低，测试结果如图 1 所示，所以将神经网络参数量化为 8 位定点数。

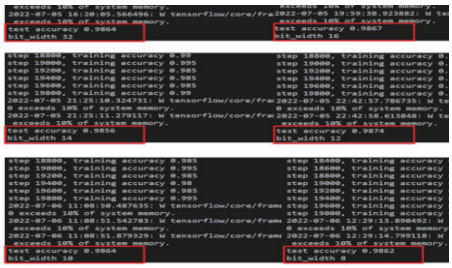

图 1 不同位数表示的神经网络参数预测的准确率

2 加速器设计

2.1 整体设计

本文设计的卷积神经网络硬件加速模块的整体结构如图 2 所示。PS 端的内存 DDR 和 ARM 接在一起，DDR 中存储配置信息，ARM 根据这些配置信息对加速器 VPU 进行控制，卷积模块、池化模块和全连接层的加速设计都在 VPU 中实现。

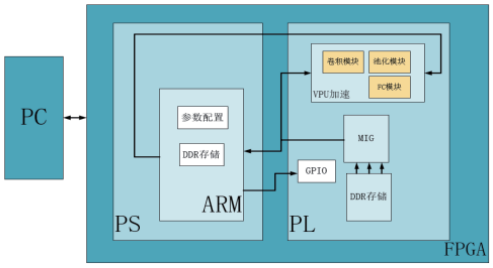

图 2 卷积神经网络硬件加速模块的整体结构

2.2 详细设计

2.2.1 数据存储方式的选择

本文使用了 NHWC 的存储格式对权重和图片的特征值进行存储，将数据先按通道方向，再按行、列依次存入 FPGA 片上缓存。这种方式在计算卷积结果的时候只需要读取卷积窗口内的数据，再经过计算传回到 DDR 中，之后计算像素点的结果直接和之前的进行拼接。使用这种方式存储的通道间数据的相关性比较弱，能更容易地实现通道间的并行计算。

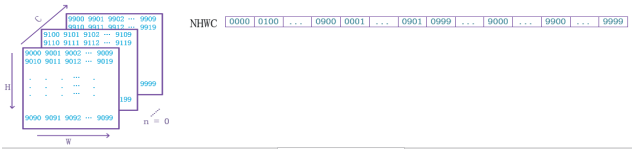

图 3 常用的两种基本的数据存储格式

2.2.2 数据复用

可以从三方面着手降低数据搬运的功耗。

一是减小数据的搬运次数，当一批数据从存储单元进入计算单元后，在不存在依赖关系的条件下让该数据执行完所有的运算，将中间运算结果暂存在寄存器单元中，这样的话就能有效地减少数据的搬运次数，该设计思想就是数据复用。数据复用具体有两种方法，分别是输入特征的复用和卷积核的复用。

二是输入特征的复用。将特征图输入进稀疏矩阵乘法模块后，同时使用 N 个卷积核和该特征图进行卷积运算。采用该方法，以特征图的卷积窗口作为最小的复用单元，将输入信息传到多个运算单元。

三是在对某一张输入图形执行卷积运算的时候，卷积核在特征图形中进行滑动，假设特征图的尺寸是卷积核尺寸的 K 倍，则对于卷积核来说进行了 K 次复用，这就是卷积核的复用。

2.2.3 卷积模块的设计

对于卷积神经网络来说，卷积运算的时序性能相当重要，对卷积层进行恰当的设计以提高卷积模块的计算效率，在很大程度上能够决定硬件加速的性能。

卷积模块包含了多个小模块，主要是预先对特征向量进行补零操作的 Padding 模块和进行乘累加的 MAC 模块，接下来将对这两个模块的详细设计进行介绍。

（1） Padding 模块的设计

为了不让特征图的大小因为卷积运算而改变，需要在运算之前对其进行 Padding 操作。Padding 操作属于对特征图进行预处理，其最重要的一步是确定输入特征图中元素的位置，通过元素的位置决定是否要对其进行补零和补零的具体位置。补零的方式有多种，本文为了降低片上资源的消耗，设计了一个专用的补零模块，数据从片上缓存传输到 MAC 模块的时候在其周围添加零元素。在读取时先产生横向和纵向两个方向的计数，决定此刻卷积窗口在哪里。当卷积窗口中央对应的数据不在特征图的边缘区域时，不需要进行补零操作；当卷积窗口中心对应的数据在特征图的边缘区域时，再根据窗口中心对应的数据是否在特征图的角落决定对该数据周围的哪些位置进行补零操作。如图 4 所示。

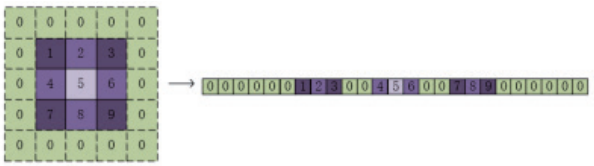

图 4 补零操作的运算示意图

（2）MAC 模块的设计

在对特征图的边缘数据进行补零之后，特征数据正式进入了卷积运算。卷积器具有按卷积核大小布置的 MAC 单元和连接在流水线中各个级别的几个移位寄存器，当输入特征图和权重数据时，在固定数量的时钟周期后输出卷积运算的结果。对 MAC 模块进行恰当的优化可以节约大量的时序资源。因此，本文的加速器使用流水线和并行乘加的方式对 MAC 模块进行优化，这节主要介绍并行乘加。

稀疏矩阵乘法中，要将输入向量和权重矩阵的所有元素相乘，再进行累加，算出当前值。如果遵循原始的方法进行累加，各个周期执行一次加法。n-1 个周期可以对 n 个数值进行累加。本文在累加操作的基础上实行了并行改进，假设有 n 个数据，最大 k 次累加，$\log_2 n \le k < \log_2 n + 1$，模块不可能有空闲的等待时间，因而有效地提高了计算资源的使用效率并节约了大量时序资源，n 越大，减少的计算时间就越多，计算效率的提升也就越明显。

2.2.4 池化模块的设计

本文使用的池化方式是最大值池化，池化窗口的维度是 2×2，在每一个池化窗口中取它的最大值作为输出的特征数据，同时设置池化步长为 2。池化后特征图的尺寸缩小，但是通道的数目并没有改变。首先按行读取数据，根据池化窗口的尺寸得到第一行的池化结果，将这个中间结果暂存起来；然后将所有行的池化结果组成一个暂存矩阵；最后进行列方向的池化，池化输入是第一步按行池化后得到的暂存矩阵，等两个方向的数据都池化结束的时候，就可以得到整个池化操作的输出特征图了。使用这种方式将二维的池化运算分解为宽度和高度两个方向上的一维池化，大大减少了计算量，同时减小了对片上存储的占用。

2.2.5 FC 层运算模块的分配

在卷积网络中，经过卷积层的运算，需要再使用全连接层提取输入数据的特征。FC 层的数据量不是最大的，但是因为它的全连通性，所以运算量是最大的。FC 层输出的计算公式为：

$$y = ReLU(Wa + b) \tag{11}$$

y 的每个元素值的计算公式为：

$$y_i = ReLU\left(\sum_{j=0}^{n-1} W_{ij} a_j + b_j\right) \tag{12}$$

3 测试结果

本设计使用的数据集是 ILSVRC 竞赛使用的数据集 ImageNet 2012，通过对数据集中的图像进行测试得到如下测试结果。第一张测试图片的编号是 ILSVRC2012_val_00049327，其对应的分类是 224 个狗类，经过测试得到该图片中的标签是 groenendael，Newfoundland，Newfoundland dog，affenpinscher，monkey pinscher，monkey dog，其测试的第一个分类标签输出了正确的结果，即狗的品种是格罗尼达尔犬。第二张测试图片的编号是 ILSVRC2012_val_00049816，其对应的分类是 423 座椅类，经过测试得到该图片中的标签是 barber chair，babershop，dining table，board，其测试的第一个分类标签同样输出了正确的结果。对神经网络经过一万次的训练，统计得出整体的预测准确率达到了 98.67%，如图 5、图 6 所示。

图 5 CNN 加速器的运算时间及识别结果

```
 Conv1 Time : 265
Pool1 Time : 9
Conv2 Time : 188
Pool2 Time : 8
FC1 Time : 1632
FC2 Time : 17
 Correct Rate:0.986700
```

图 6 CNN 加速器的预测准确率

根据测试结果得到 CNN 加速器的推理时间大致是 168 ms，以第二张测试图片的测试时间为 168.751 ms 进行计算，根据 VGG-16 神经网络的 OPS 为 30.94 GB 可以计算出本文的卷积神经网络加速器的算量是 183.35 GOPS，时钟频率为 150 MHz，总功耗是 5.61 W，如图 7 所示。

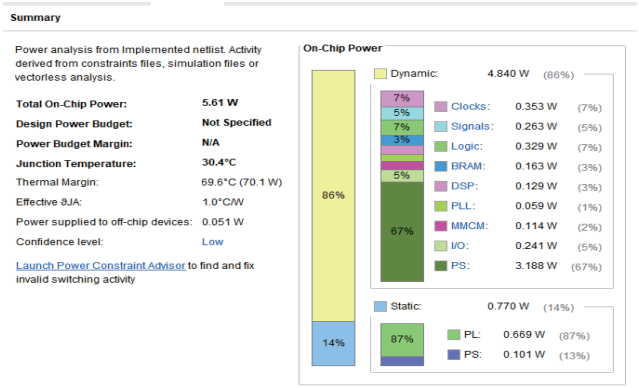

图 7 CNN 加速器的功耗图

使用硬件平台的计算速度除以总消耗的功耗，可以获得能量效率。本文设计的 CNN 网络加速器的能量效率为 32.68 GOPS/W。本文也与 CPU[11] 平台和 GPU 平台上执行相同的神经网络模型进行对比，如表 1 所示。在型号是 Intel I7-4790K 的 CPU 上的推理时间为 31 600 ms。GPU 的型号是 NVIDIA TX2，其运行同样的网络需要的时间是 46.5 ms，但是其功耗较大，是 17.3 W，导致其能量效率不高，只有 19.25 GOPS/W。

表 1 本文 CNN 加速器和 CPU 和 GPU 平台的对比

平台	CPU	GPU	FPGA
型号	Intell7-4790K	NVIDIA TX2	ZCU102
功耗 /W	—	17.3	5.61
时间 /ms	31 600	46.5	168.75
算力 /GOPS	0.49	333	183.35
能效 /(GOPS·W^{-1})	—	19.25	32.68

4 总结

本文针对压缩后的卷积神经网络模型，设计了一个基于 Xilinx 的 ZCU102 芯片的 CNN 神经网络加速框架，分别从数据的存储方式、时序资源的节约方法数据复用（特征数据的复用和卷积核的复用）、卷积模块的设计、池化层的设计和全连接层的设计等方面对加速器系统做出了详细的介绍。最后，通过各个模块的仿真波形证明了模块功能的正确性，测试了系统的运行时间，分析了资源占用情况和功耗，得到系统的具体性能和加速效果，对比了本文设计的 CNN 加速器和 GPU、CPU 三种硬件平台的性能。结果显示，本文设计的卷积神经网络加速器在能量效率方面是 GPU 的 1.70 倍，在计算速度方面是 CPU 的 374.18 倍，和其他 FPGA 平台上的加速器相比在性能方面有更大的优势，达到了一个较满意的效果。

参考文献：

[1] MCCULLOCH W S, PITTS W. A logical calculus of the ideas immanent in nervous activity[J]. The bulletin of mathematical biophysics,1943,5(4):115-133.

[2] 黎亚雄，张坚强，潘登，等．基于 RNN-RBM 语言模型的语音识别研究 [J]. 计算机研究与发展，2014, 51(9): 1936-1944.

[3] 朱小燕，王昱，徐伟．基于循环神经网络的语音识别模型 [J]. 计算机学报，2001(2):213-218.

[4] 张剑，屈丹，李真．基于词向量特征的循环神经网络语言模型 [J]. 模式识别与人工智能，2015,28(4):299-305.

[5] HOPFIELD J J. Neural networks and physical systems with emergent collective computational abilities[J]. Proceedings of the national academy of sciences of the United States of America, 1982,79(8):2554-2558.

[6] HOCHREITER S,SCHMIDHUBER J. Long short-term memory[J]. Neural computation,1997,9(8):1735-1780.

[7] JIANG F, FU Y, GUPTA B B, et al. Deep learning based multi-channel intelligent attack detection for data security[J]. IEEE transactions on sustainable computing, 2020,5(2):204-212.

[8] KARNIN E D. A simple procedure for pruning back-propagation trained neural networks[J]. IEEE transactions on neural networks,1990,1(2):239-342.

[9] ZHOU S, WANG Y Z, HE W, et al. Balanced quantization: an effective and efficient approach to quantized neural networks[J]. Journal of computer science and technology, 2017, 32(4): 37-43.

[10] 张奕玮．基于 FPGA 的高能效比 LSTM 预测算法加速器的设计与实现 [D]. 合肥：中国科学技术大学，2018.

[11] 郑俊伟．基于 FPGA 的卷积神经网络并行加速技术研究 [D]. 西安：西安电子科技大学，2021.

【作者简介】

薛慧敏（1998—），女，陕西咸阳人，硕士研究生，助理工程师，研究方向：神经网络。

（收稿日期：2024-03-25）