CN 43-1258/TP
ISSN 1007-130X

计算机工程与科学
Computer Engineering & Science

第 46 卷第 9 期 2024 年 9 月
Vol. 46, No. 9, Sep. 2024

文章编号:1007-130X(2024)09-1547-07

适应于硬件部署的神经网络剪枝量化算法*

王　鹏[1,2],张嘉诚[1],范毓洋[1,2]

(1.中国民航大学民航航空器适航审定技术重点实验室,天津 300399;
2.中国民航大学安全科学与工程学院,天津 300399)

摘　要:深度神经网络由于性能优异已经在图像识别、目标检测等领域广泛应用,然而其包含大量参数和巨大计算量,导致在需要低延时和低功耗的移动边缘端部署时困难。针对该问题,提出一种用移位加法代替乘法运算的压缩算法,通过对神经网络进行剪枝和量化将参数压缩至低比特。该算法在乘法资源有限的情况下降低了硬件部署难度,可满足移动边缘端低延时和低功耗的要求,提高运行效率。对 ImageNet 数据集经典神经网络进行了实验,结果表明神经网络的参数在压缩到 4 bit 的情况下,其准确率与全精度神经网络的基本一致,甚至在 ResNet18、ResNet50 和 GoogleNet 网络上的 Top-1/Top-5 准确率还分别提升了 0.38%/0.22%,0.35%/0.21%和 1.14%/0.57%。对 VGG16 第 8 层卷积层进行实验,将其部署在 Zynq7035 上,结果表明,压缩后的网络在使用的 DSP 资源减少 43%的情况下缩短了 51.1%的推理时间,并且减少了 46.7%的功耗。

关键词:深度神经网络;硬件;剪枝;量化;FPGA

中图分类号:TP183　　**文献标志码**:A

doi:10.3969/j.issn.1007-130X.2024.09.004

A neural network pruning and quantization algorithm for hardware deployment

WANG Peng[1,2], ZHANG Jia-cheng[1], FAN Yu-yang[1,2]

(1. Key Laboratory of Civil Aircraft Airworthiness Technology, Civil Aviation University of China, Tianjin 300399;
2. College of Safety Science and Engineering, Civil Aviation University of China, Tianjin 300399, China)

Abstract: Due to their superior performance, deep neural networks have been widely applied in fields such as image recognition and object detection. However, they contain a large number of parameters and require immense computational power, posing challenges for deployment on mobile edge devices that require low latency and low power consumption. To address this issue, a compression algorithm that replaces multiplication operations with bit-shifting and addition is proposed. This algorithm compresses neural network parameters to low bit-widths through pruning and quantization. This algorithm reduces the hardware deployment difficulty under limited multiplication resources, meets the requirements of low latency and low power consumption on mobile edge devices, and improves operational efficiency. Experiments conducted on classical neural networks with the ImageNet dataset revealed that when the neural network parameters were compressed to 4 bits, the accuracy remained essentially unchanged compared to the full-precision neural network. Furthermore, for ResNet18, ResNet50, and GoogleNet, the Top-1/Top-5 accuracies even improved by 0.38%/0.22%, 0.35%/0.21%, and 1.14%/0.57%, respectively. When testing the eighth convolutional layer of VGG16 deployed on Zynq7035, the results showed that the compressed network reduced the inference time by 51.1% and power consumption by 46.7%, while using 43% fewer DSP resources.

* **收稿日期**:2022-07-18;**修回日期**:2023-05-22
基金项目:国家重点研发计划(2021YFB1600600)
通信地址:300399 天津市中国民航大学安全科学与工程学院
Address: College of Safety Science and Engineering, Civil Aviation University of China, Tianjin 300399, P. R. China

Key words:deep neural networks;hardware;pruning;quantization;FPGA

1 引言

近十年来,得益于计算机算力的不断提升和大数据时代的到来,深度神经网络技术得到了飞速的发展,在自然语言处理[1]、图像识别[2]等领域表现出了优秀的性能。但是,优秀的性能伴随着参数量和计算量的急剧增加。以经典图像识别 ImageNet 数据集为例,针对该数据集提出的神经网络模型深度逐年增加[2-5],大量的参数和计算量导致神经网络模型难以部署到要求实时性和低功耗的移动边缘端。

通过知识蒸馏和改进卷积核,可以优化神经网络结构。Hinton 等人[6]通过知识蒸馏训练小型学生网络。MobileNet[7]使用深度可分离卷积减少了参数量。ShuffleNet[8]使用分组卷积减少了卷积核个数和计算量。SVD(Singular-Value Decomposition)[9,10]将高维权重矩阵分解成低维矩阵,压缩了参数量。剪枝将神经网络部分权重变成零。阮晓峰等人[11]提出了一种基于模型特征学习增强的动态剪枝。Han 等人[12]根据每一层权重大小设置阈值,从而剪枝部分权重,并通过重新训练恢复精度。Choudhary 等人[13]使用贝叶斯优化对每一卷积层进行过滤器级别剪枝,加速了神经网络在边缘端的推理过程。Salehinejad 等人[14]提出了一种基于能量的暂时隐去(dropout)剪枝方法,实现了对参数超过 50% 的剪枝比例。但是,在上述方法中,依然使用浮点数表示参数,难以在存储资源有限的边缘端部署。

量化将神经网络权重由高比特变成低比特。Chen 等人[15]通过对多个网络进行不同比特的压缩实验解释了量化是一种特殊的正则化方法。张帆等人[16]提出基于损失最小化的因子动态舍入方法。XNOR-Net[17]是一种极端的量化方式,其将原始网络的权重进行二值化。Wu 等人[18]对原始网络在矩阵级别使用了 k-means 量化。LQ-Nets(Learned Quantization-Nets)[19]在训练过程中对权重和激活值进行量化,将其分解成不同的数相加,然后再进行编码保存。Han 等人[20]对原始网络首先进行剪枝,然后再进行量化,最后用霍夫曼编码进行进一步压缩。Li 等人[21]在量化的同时考虑了鲁棒性,提出非敏感扰动损失函数,加强了量化网络的鲁棒性。Jung 等人[22]提出了一种在精度受限情况下最具容错性的等效网络优化方法,利用自适应舍入偏移控制进行量化。通过量化技术可以有效减少存储量,但是在进行前向推理计算时依然需要进行浮点乘法,所需运算资源多,难以在需要低延时、低功耗的边缘端部署。

上述神经网络压缩算法中,知识蒸馏、改进卷积核和剪枝减少了参数量,但是参数依然是用高比特存储。量化将参数变成低比特,但是依然需要使用浮点乘法。研究表明,神经网络前向推理过程需要使用大量浮点乘法运算单元。例如,李沙沙等人[23]利用 FPGA(Field Programmable Gate Array)设计的通用卷积神经网络加速器中,DSP(Digital Signal Processing)资源的占有率达到了 99.40%,而 LUT(Look_Up_Table)资源的占有率仅为 46.79%,可见 DSP 资源限制了神经网络的加速效果。为了在 DSP 资源有限的情况下减少运算量,本文提出一种适应于硬件部署的神经网络压缩算法,将权重用 2^n 相加减的形式表示,用移位加法代替浮点乘法,在面积受限情况下提高并行度,从而降低延时和功耗。本文首先对神经网络根据每层权重分布特点剪枝,再进行迭代量化逐步降低量化损失,并通过重新训练可以恢复甚至提高准确率。

2 压缩算法

本文提出的压缩算法首先对神经网络进行适应性剪枝,并通过重新训练恢复精度。然后通过迭代量化依次对权重进行量化,量化时使用改进 k-means 聚类将权重变成 2^n 相加减形式,每一次量化后都重新训练。算法流程图如图 1 所示。

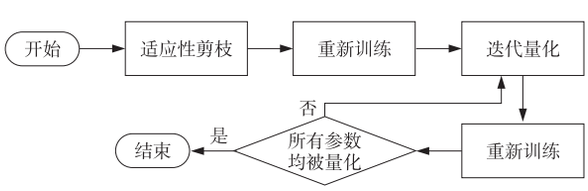

Figure 1 Flowchart of the proposed algorithm

图 1 本文算法流程

2.1 适应性剪枝

剪枝通常通过设置数值阈值或者比例阈值的方式使绝对值较小的权重变成 0,但是这 2 种剪枝方式无法适应不同卷积层或者全连接层的权重分布特点。若权重分布集中在 0 附近,按照数值阈值的方式会剪掉过多权重,精度下降明显;若权重分布集中在距离 0 较远处,按照比例阈值的方式会剪

掉部分重要权重,精度下降明显。本文使用的适应性剪枝按照每一层权重最大绝对值的比例剪枝,这种剪枝方式考虑了不同层的权重分布特点,并且通过重新训练恢复甚至提升了准确率。无论权重如何分布都可以根据权重最大绝对值剪掉不重要权重,从而在剪枝过程中控制精度下降。图 2 给出了 ResNet18 网络中不同层的权重分布,可以观察到图 2a 中权重分布在绝对值较小范围内,而图 2b 权重分布更为均匀。适应性剪枝将考虑到不同层分布特点差别进行剪枝。

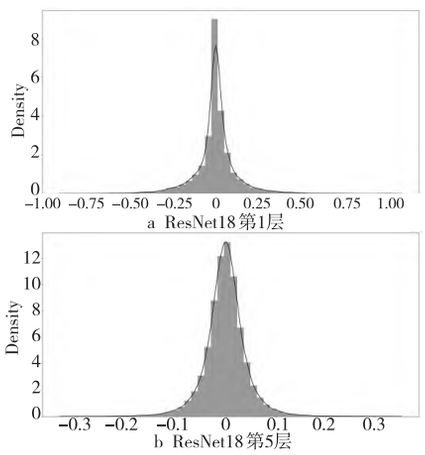

Figure 2　Kernel density of different layer weights

图 2　不同层权重核密度

适应性剪枝流程如图 3 所示,首先根据每一层权重绝对值的最大值进行剪枝,绝对值小于该值乘以剪枝因子的权重将被剪掉,然后通过重新训练恢复精度。剪枝因子是超参数,其值的选取将在 3.4 节进一步讨论。这种剪枝方式适应了不同层的权重分布。在某些网络的一些层中会出现极个别的较大值,此时剪枝参考值选择为权重绝对值最大值的 0.99 倍,选定剪枝参考值后剪枝每一层绝对值小于剪枝参考值乘以剪枝因子的权重。

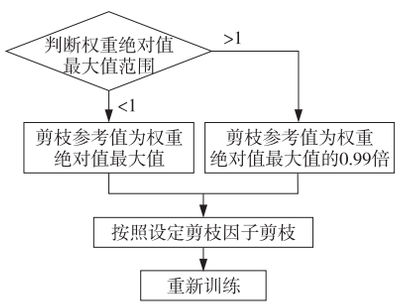

Figure 3　Flowchart of adaptive pruning

图 3　适应性剪枝流程图

2.2　迭代量化

传统的量化方法通常一次性将神经网络所有的权重压缩到低比特,导致精度损失较大,并且难以通过重新训练恢复精度。本文提出的量化方法通过逐步迭代量化减小了精度损失。首先按照每一层的权重绝对值大小进行分类,绝对值较大的优先进行量化,之后按照分类量化顺序依次进行量化;每一次量化后都会通过重新训练恢复准确率,且量化后的权重在接下来的训练中不发生改变。量化初期对绝对值较大权重进行量化,由于该类权重在卷积计算中占有较大比重,因此量化对精度造成较大损失,由于未量化权重占比高,重新训练可调整参数多,因此可以通过重新训练恢复精度。量化后期对绝对值较小权重进行量化,该类权重在卷积核计算中影响较小,因此量化后准确率损失小,通过对剩下权重进行训练即可恢复精度。每一次量化后准确率的下降情况如图 4 所示,图 4a 代表 ResNet18 网络,图 4b 代表 AlexNet 网络。由于第 1 次量化针对绝对值较大的权重,因此量化后准确率下降明显,之后的量化针对绝对值较小的权重,量化后准确率下降较小。每一次重新训练后准确率的提升情况如图 5 所示,图 5a 代表 ResNet18 网路,图 5b 代表 AlexNet 网路。在量化初期通过调整大部分未被量化权重可以较好地恢复精度,量化后期未被量化权重少,恢复的精度较小。图 4 和图 5 说明迭代量化具有“互补特性”,量化初期精度下降明显,通过重新训练恢复精度的能力较强;量化后期精度下降较小,通过重新训练恢复精度的能力较弱。最终通过迭代量化使得压缩后的神经网络的准确率与原始神经网络的保持一致甚至还有提升。

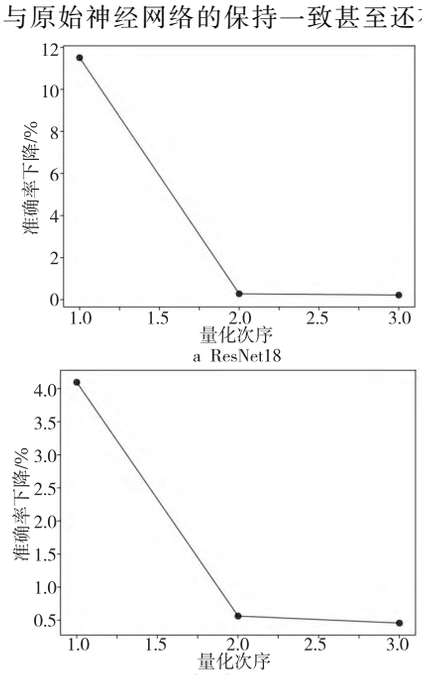

Figure 4　Accuracy decreasement of each quantization

图 4　每次量化的精度下降

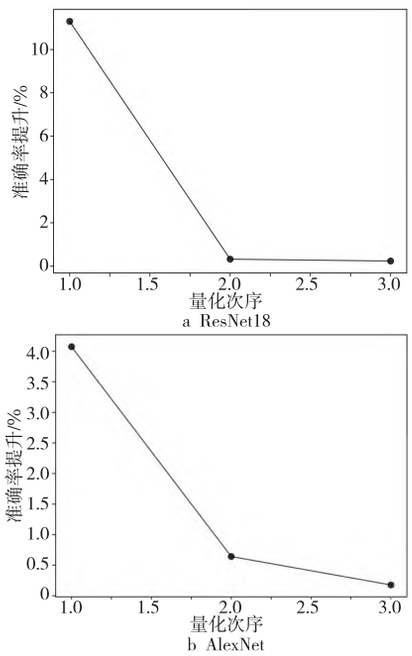

Figure 5 Accuracy increasement of each train

图 5 每次重新训练后的精度提升

2.3 改进 k-means 聚类

为了便于在硬件中部署神经网络,降低计算量,本文针对传统 k-means 聚类进行改进,将所有权重用 2^n 加减表示,进而用加法移位操作代替乘法。传统 k-means 聚类最小化公式如式(1)所示:

$$J(\boldsymbol{c},\boldsymbol{\theta})=\sum_{i=1}^{M}\|w_i-\theta_{c_i}\| \tag{1}$$

其中,w_i 表示第 i 个样本,M 表示权重总数,$\boldsymbol{\theta}$ 表示聚类中心,c_i 表示第 i 个样本所属的簇,将 M 个权重聚类到 K 簇。本文使用改进 k-means 聚类,将聚类后的值分解成 2^n 相加减,聚类后的权重在进行神经网络推理时通过加法和移位操作完成,可以有效降低延时和功耗,便于在硬件上部署神经网络。

通过式(2)和式(3)将 K 个聚类中心变成 2^n 相加减,首先确定 N,如式(3)所示:

$$\begin{cases}N_{\text{floor}}=floor(\text{lb}\ p)\\N_{\text{ceil}}=ceil(\text{lb}\ p)\end{cases} \tag{2}$$

$$N=\begin{cases}N_{\text{floor}},\text{if}\,|2^{N_{\text{floor}}}-p|<|2^{N_{\text{ceil}}}-p|\\N_{\text{ceil}},\text{others}\end{cases} \tag{3}$$

其中,p 表示该次量化权重中绝对值最大值,$floor(\cdot)$ 表示向下取整,$ceil(\cdot)$ 表示向上取整。上述公式中首先根据该次量化中绝对值最大的权重求出 2 个可能的值 N_{floor} 和 N_{ceil};其次根据 N_{floor} 和 N_{ceil} 的二次幂到 p 的距离确定最终的 N 值;最后求解出由 $[\pm 2^N,\pm 2^{N-1},\cdots,\pm 2^{N-m}]$ 相加减所组成的集合,并将每一个聚类中心变换为上述集合中离其最近的元素,记为 $\bar{\theta}_i$,$i\in[1,K]$。m 是一个超参数,m 越大,转换为 2^n 的精度越高,m 越小,转换为 2^n 的精度越低。在硬件中部署神经网络时 N 代表了移位次数,m 代表了加法次数。经过上述转换后得到最终聚类中心 $\bar{\theta}_i$,再将权重量化到聚类中心。

2.4 算法框架

图 6 给出了迭代量化的简单图示,图中首先将权重分成 2 部分,将第 1 部分权重使用 k-means 聚类之后再将聚类中心转换成 2^n 相加减,重新训练第 2 部分权重,再对其进行 k-means 聚类并转换聚类中心。总共需要 6 个聚类中心,因此每个权重只需要 3 bit 存储。在完成聚类中心转换之后所有的权重都变成了 2^n 相加减的形式,便于在硬件上使用移位相加操作部署神经网络。

整体算法流程如算法 1 所示,首先进行适应性剪枝,再对权重进行分类,之后每一类分别进行改进 k-means 聚类并重新训练未被量化权重,直到所有权重都被量化。

算法 1 适应剪枝量化压缩算法

输入:训练后全精度网络。

输出:经剪枝量化后的低比特存储网络。

Step 1 进行自适应剪枝;

Step 2 重新训练恢复精度;

Step 3 按照权重绝对值大小划分权重量化顺序;

Step 4 开始迭代量化:

Step 4.1 重置学习率等参数;

Step 4.2 每层使用改进 k-means 聚类;

Step 4.3 重新训练剩下参数;

Step 5 结束。

3 实验与结果分析

3.1 实验过程

本文使用的 ImageNet 数据集是目前在图像识别领域最具有影响力的数据集之一,该数据集包含 120 万幅训练图像和 5 万幅测试图像,总共包含 1 000 个类别。将本文的剪枝量化算法应用于 AlexNet、VGG16、GoogleNet、ResNet18 和 ResNet50 网络架构,使用压缩后的网络在测试集上的 Top-1 准确率和 Top-5 准确率作为性能指标。同时,使用 Pytorch 框架,将所有压缩后的网络与压

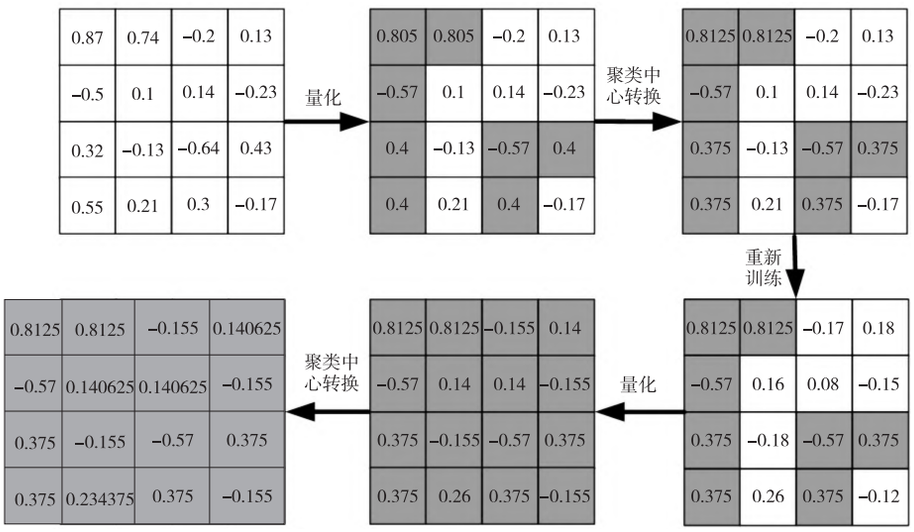

Figure 6　Iterative quantization

图 6　迭代量化

缩前 32 bit 网路进行对比。

3.2　参数设置

实验中所有模型均使用 SGD(Stochastic Gradient Descent)优化器,所有模型在进行 k-means 聚类中心转换时采用相同精度,m 设置为 4。所有模型每一次剪枝和量化后更新学习率,重新训练 4 次,每训练 2 次学习率变为 1/10。设置的参数如表 1 所示。

3.3　实验结果

使用本文压缩算法后的网络的 Top-1 准确率和 Top-5 准确率如表 2 所示。从表 2 可以看到,当把原始网络从 32 bit 权重压缩到 4 bit 权重时 GoogleNet、ResNet18 和 ResNet50 网络的 Top-1 准确率和 Top-5 准确率均有一定程度的提升,这说明在考虑了不同层权重分布特点的情况下,对每一层进行有针对性的适应性剪枝和按照权重绝对值设置迭代量化的策略有助于提升准确率。此外,所有的权重变成了 2^n 相加减的形式,便于在硬件上部署神经网络。

Table 2　Comparison of the accuracy between the compressed and the original networks

表 2　压缩后的网络与原网络准确率对比

网络	位宽	Top-1 准确率/%	Top-5 准确率/%
AlexNet ref	32	56.522	79.066
AlexNet	4	56.070	78.792
VGG16 ref	32	71.592	90.382
VGG16	4	71.508	90.382
GoogleNet ref	32	69.778	89.530
GoogleNet	4	70.918	90.100
ResNet18 ref	32	69.758	89.076
ResNet18	4	70.142	89.300
ResNet50 ref	32	76.130	92.862
ResNet50	4	76.480	93.072

表 3 给出了 AlexNet 使用本文压缩算法和其他压缩算法时的准确率和压缩比。从表 3 中可以看出,使用本文压缩算法相较于其他压缩算法在压缩比较高的情况下准确率相较原始网络下降较少。

Table 1　Parameter settings

表 1　参数设置

网络	批次大小	动量	权值衰减	初始学习率	量化次序	剪枝因子
AlexNet	256	0.9	0.000 5	0.001 0	[1,0.65,0.35]	0.005
VGG16	32	0.9	0.000 5	0.000 1	[1,0.50,0.25]	0.010
GoogleNet	80	0.9	0.000 2	0.000 1	[1,0.70,0.40]	0.010
ResNet18	80	0.9	0.000 5	0.000 1	[1,0.50,0.25]	0.010
ResNet50	32	0.9	0.000 5	0.000 1	[1,0.50,0.25]	0.010

Table 3 Accuracy and compression rate of AlexNet under various compression algorithms

表 3 各种压缩算法下 AlexNet 的准确率和压缩率

算法	Top-1 准确率/%	Top-5 准确率/%	*Compress Rate*
Baseline Model[2]	56.522	79.066	1×
Edropout[14]	56.030	77.920	1.2×
SVD[10]	55.980	79.440	5×
NO+ARO[22]	53.735	-	5.3×
Q-CNN[18]	55.142	78.226	15.4×
Adaptive pruning+ Iterative quantization	**56.070**	**78.792**	**8×**

3.4 剪枝的影响

迭代量化中根据设定好的比例对未剪枝的参数进行分类后聚类,若剪枝因子选择过大,则在迭代量化中每一类权重数量都减少,进行聚类时在聚类中心数量不变的情况下聚类损失少,此时由剪枝带来的准确率损失较大,由量化带来的准确率损失较小。若剪枝因子选择过小,则在迭代量化中每一类权重数量都增加,进行聚类时在聚类中心数量不变的情况下聚类损失大,此时由剪枝带来的准确率损失较小,由量化带来的准确率损失较大。因此,存在一个剪枝因子的中间值使得最终由剪枝和量化整体导致的准确率损失最小。本文对 ResNet18 网络在不进行适应性剪枝和在选择不同剪枝因子的情况下进行了实验,实验结果如表 4 所示,当剪枝因子选择为 0.010 时,Top-1 准确率和 Top-5 准确率高于剪枝因子为 0.005 和 0.015 时的结果。实验结果说明,剪枝因子存在一个中间值,使得压缩后神经网络的准确率最高,针对不同的网络可以通过设置不同剪枝因子进行实验来确定最优剪枝因子。

Table 4 Comparison of the accuracy of the compressed and the original networks under different pruning factors

表 4 不同剪枝因子下压缩后的网络与原网络准确率对比

剪枝因子	Top-1 准确率/%	Top-5 准确率/%
0.005	70.002	89.336
0.010	70.142	89.300
0.015	69.946	89.192

3.5 硬件部署结果

针对压缩前和压缩后的 VGG16 第 8 层卷积层进行硬件部署,该卷积层输入为 256 通道的 28×28 特征图,输出为 512 通道的 28×28 特征图。通过 HLS(High-Level Synthesis)分别部署在 Xilinx 的 Zynq7035 板卡上,将卷积层输出通道并行展开计算,压缩前并行展开度为 32,压缩后并行展开度为 64,时钟频率为 130 MHz,资源占用情况如表 5 所示,推理时间和功耗如表 6 所示。从表 5 可以看出,压缩前的卷积层使用了 73%的 DSP 资源(卷积层大量的浮点运算使其需要使用大量的 DSP 资源)。而压缩后的卷积层将浮点运算转换成使用移位加法的定点运算,使用的 DSP 资源减少了 43%,FF 资源减少了 14%,从而更易于在硬件资源有限的边缘端部署神经网络。从表 6 可以看出,压缩后的卷积层相较压缩前推理时间减少了 51.1%,功耗减少了 46.7%。使用移位加法的定点运算相比浮点运算具有更低的延时,将权重从 32 bit 压缩到 4 bit,加快了从内存中读取权重的速度,并降低了读取权重的功耗,从而更有利于满足边缘端低延时和低功耗的要求。

Table 5 Resource consumption

表 5 资源占用情况

资源类型	总资源	压缩前使用率/%	压缩后使用率/%
DSP	900	73	30
FF	343 800	29	15
LUT	171 900	35	30
LUTRAM	70 400	13	16
BRAM	500	20	22

Table 6 Results of inference time and power consumption

表 6 推理时间和功耗结果

	推理时间/μs	功耗/W
压缩前	58 825	5.05
压缩后	28 772	2.69

4 结束语

神经网络具有大量的参数和计算量,因此难以在移动端部署。本文提出的压缩算法对神经网络实现了 8× 的压缩效果,并将所有权重变成 2^n 加减形式,用移位加法代替浮点乘法,使得神经网络易于在硬件上部署,减少了部署时的存储量,降低了推理时的吞吐量、延迟和功耗。通过对神经网络每一层权重进行分析、自适应剪枝和迭代量化,在压缩神经网络模型的同时控制了精度下降。在 ImageNet 数据集上的实验结果说明,使用该压缩算法,神经网络的准确率基本不变甚至还有所提升。在硬件上部署的结果证实了压缩后的卷积层在使用更少硬件资源的情况下实现了更低的延时和功耗。

参考文献:

[1] Vaswani A,Shazeer N,Parmar N,et al. Attention is all you

need[C]//Proc of the 31st International Conference on Neural Information Processing Systems,2017:6000-6010.

[2] Krizhevsky A,Sutskever I,Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM,2017,60(6):84-90.

[3] Szegedy C,Liu W,Jia Y Q,et al. Going deeper with convolutions[C]//Proc of 2015 IEEE Conference on Computer Vision and Pattern Recognition,2015:1-9.

[4] Simonyan K,Zisserman A. Very deep convolutional networks for large-scale image recognition[C]//Proc of International Conference on Learning Representations,2015:1-14.

[5] He K M,Zhang X Y,Ren S Q,et al. Deep residual learning for image recognition[C]//Proc of 2016 IEEE Conference on Computer Vision and Pattern Recognition,2016:770-778.

[6] Hinton G,Vinyals O,Dean J. Distilling the knowledge in a neural network[J]. arXiv:1503.02531,2015.

[7] Howard A G, Zhu M L,Chen B,et al. MobileNets:Efficient convolutional neural networks for mobile vision applications [J]. arXiv:1704.04861,2017.

[8] Zhang X Y,Zhou X Y,Lin M X,et al. ShuffleNet:An extremely efficient convolutional neural network for mobile devices[C]//Proc of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,2018:6848-6856.

[9] Gong Y C,Liu L,Yang M,et al. Compressing deep convolutional networks using vector quantization[J]. arXiv:1412.6115,2014.

[10] Denton E,Zaremba W,Bruna J,et al. Exploiting linear structure within convolutional networks for efficient evaluation[C]//Proc of the 27th International Conference on Neural Information Processing Systems,2014,1:1269-1277.

[11] 阮晓峰,胡卫明,刘雨帆,等. 基于动态稀疏和特征学习增强的模型剪枝[J]. 中国科学(技术科学),2022,52(5):667-681.
Ruan Xiao-feng,Hu Wei-ming,Liu Yu-fan,et al. Dynamic sparsity and model feature learning enhanced training for convolutional neural network-pruning[J]. SCIENTIA SINICA Technologica,2022,52(5):667-681.

[12] Han S,Pool J,Tran J,et al. Learning both weights and connections for efficient neural network[C]//Proc of the 28th International Conference on Neural Information Processing Systems,2015:1135-1143.

[13] Choudhary T,Mishra V,Goswami A,et al. Inference-aware convolutional neural network pruning[J]. Future Generation Computer Systems,2022,135(C):44-56.

[14] Salehinejad H,Valaee S. EDropout:Energy-based dropout and pruning of deep neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022,33(10):5279-5292.

[15] Chen W T,Qiu H L,Zhuang J,et al. Quantization of deep neural networks for accurate edge computing[J]. ACM Journal on Emerging Technologies in Computing Systems,2021,17(4),Article No.:54.

[16] 张帆,黄赟,方子茁,等. 卷积神经网络的损失最小训练后参数量化方法[J]. 通信学报,2022,43(4):114-122.
Zhang Fan,Huang Yun,Fang Zi-zhuo,et al. Lost-minimum post-training parameter quantization method for convolutional neutral network[J]. Journal on Communications,2022,43(4):114-122.

[17] Rastegari M, Ordonez V, Redmon J,et al. XNOR-Net:ImageNet classification using binary convolutional neural networks[J]. arXiv:1603.05279,2016.

[18] Wu J X,Leng C,Wang Y H,et al. Quantized convolutional neural networks for mobile devices[C]//Proc of 2016 IEEE Conference on Computer Vision and Pattern Recognition,2016:4820-4828.

[19] Zhang D Q, Yang J L,Ye D Q,et al. LQ-Nets:Learned quantization for highly accurate and compact deep neural networks[J]. arXiv:1807.10029,2018.

[20] Han S, Mao H Z,Dally W J. Deep compression:Compressing deep neural networks with pruning,trained quantization and huffman coding[C]//Proc of the 4th International Conference on Learning Representations,2016:1-14.

[21] Li X B,Jiang H X,Huang S X,et al. Robustness-aware 2-bit quantization with real-time performance for neural network [J]. Neurocomputing, 2021,455:12-22.

[22] Jung Y, Kim H, Choi Y,et al. Quantization-error-robust deep neural network for embedded accelerators[J]. IEEE Transactions on Circuits and Systems II:Express Briefs,2022,69(2):609-613.

[23] 李沙沙,李夏禹,刘珊珊,等. 一种基于FPGA的通用卷积神经网络加速器的设计与实现[J]. 复旦学报(自然科学版),2022,61(1):69-76.
Li Sha-sha,Li Xia-yu,Liu Shan-shan,et al. Design and implementation of an FPGA-based general accelerator for convolution neutral networks[J]. Journal of Fudan University (Natural Science),2022,61(1):69-76.

作者简介:

王鹏(1982-),男,新疆霍城人,博士,研究员,研究方向为民机系统安全性设计与评估和机载电子硬件适航技术。**E-mail**:pwang_cauc@163.com

WANG Peng, born in 1982, PhD, research fellow, his research interests include aircraft system design & evaluation of safety and electronic hardware airworthiness technology.

张嘉诚(1999-),男,湖北武汉人,硕士生,研究方向为机载电子适航设计验证和机载电子软错误防护。**E-mail**:895203099@qq.com

ZHANG Jia-cheng, born in 1999, MS candidate, his research interests include airborne electronic airworthiness design & verification and airborne electronic soft error protection.