

引用格式:王旭,王晶,张伟功. 基于神经元容错度分析的神经网络裁剪与近似计算技术研究[J]. 微电子学与计算机, 2019, 36(11):70-75. [WANG X, WANG J, ZHANG W G. Research on neural network pruning and approximate computing technology based on neuron fault tolerance analysis [J]. Microelectronics & Computer, 2019, 36(11):70-75]

基于神经元容错度分析的神经网络裁剪与近似计算技术研究

王 旭¹, 王 晶^{1,2}, 张伟功^{1,3}

(1 首都师范大学 信息工程学院, 北京 100048; 2 中国科学院计算技术研究所 体系结构重点实验室, 北京 100086;
3 电子系统可靠性北京市重点实验室, 北京 100048)

摘 要: 本文将神经元裁剪和近似计算技术相结合, 首先提出基于统计排序的神经元容错能力量化方法. 然后, 为了识别神经元的裁剪度, 根据神经元的容错能力提出神经元重要程度排序算法. 其次, 引入轻量级的重训练, 提出循环裁剪法, 以探寻最优裁剪率. 最后, 根据神经元的容错能力, 在神经网络运行过程中使用近似计算技术进一步降低功耗开销. 本文通过两个实验, 证明了该技术的有效性, 其中以 MNIST 为例, 在精度损失小于 5% 的情况下, 压缩率达到 50%, 节能 1.35 倍.

关键词: 神经网络; 神经元容错能力; 节点裁剪; 近似计算

中图分类号: TP319

文献标识码: A

文章编号: 1000—7180(2019)11—0070—06

Research on neural network pruning and approximate computing technology based on neuron fault tolerance analysis

WANG Xu¹, WANG Jing^{1,2}, ZHANG Wei-gong^{1,3}

(1 Capital Normal University Information Engineering College, Beijing 100048, China;
2 Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100086, China;
3 Beijing Key Laboratory of Electronic System Reliability and Prognostics,
Capital Normal University, Beijing 100048, China)

Abstract: This paper proposes to use neuron node pruning and approximate computing simultaneously. First, we propose a method to quantify the fault tolerance capability of neurons based on statistics. Then, to identify whether the neuron can be pruned, an importance ranking algorithm is proposed based on the fault tolerance capability. Next, introducing retrain and cyclic pruning to find the optimal pruning rate. Finally, approximate computing technique is used to further reduce power consumption during neuron network execution. The effectiveness of above technique is proved by two experiments. In the case of MNIST dataset, the compression rate is 50% and the power saving is 1.35× when the output accuracy loss is less than 5%.

Key words: neural network; neuron fault tolerance capability; node pruning; approximate computing

1 引言

近年来, 不断发展的神经网络结构也验证了更

深的层级和参数可以有效的提高神经网络输出精度, 但是所需硬件资源逐渐增多、消耗的能源增加. 因此需要有效的方法对神经网络规模进行压缩, 并

收稿日期: 2019—02—20; 修回日期: 2019—03—11

基金项目: 国家自然科学基金(61772350, 61741211); 北京市高水平教师队伍建设计划项目(CIT&TCD201704082, CIT&TCD20170322); 体系结构国家重点实验室开放项目(CARCH201607); 北京市科技新星计划项目(Z181100006218093); 未来芯片高精尖中心开放课题(KYJJ2018008)

优化其对资源的消耗。

神经网络裁剪是现今流行的压缩方法之一,其通过压缩神经网络参数量达到降低资源开销的目的。权重裁剪^[1]和神经元裁剪(节点裁剪)^[2]更多的关注于缩减神经网络尺寸,而不是能耗的优化^[3]。近似计算技术利用人类感知能力的限制或应用本身对含噪音的输入数据的固有容错能力,降低对所有计算都要达到100%精度的要求,产生可以接受的近似结果^[4]。在神经网络运行过程中引入近似计算技术可以优化其对能源的消耗,而不会改变神经网络的尺寸。然而,近似计算技术可能会引起电路可靠性降低,很可能严重危害到神经网络的精度和性能。所以,我们需要在压缩神经网络尺寸、降低能耗的同时尽量维持原有的输出精度和性能的有效方法。

因为神经网络有天然的容错特性,裁剪和近似技术可能不会造成输出精度快速下降。更重要的,神经网络中不同神经元的容错能力也是不同的,容错能力越高,神经元可以容忍的偏差就越大。因此,为了解决上述问题,本文结合神经元的容错特性,提出将神经元裁剪和近似计算技术相结合。

2 相关研究

随着近似计算的发展,人工神经网络的近似方法遍及软件级和电路级。

(1)软件级。神经网络涉及大量的计算和访存操作,文献[4]使用精度扩缩技术减少输入操作数和网络权重的位宽,在引入轻微的输出结果偏差的情况下有效的提高了性能并节省了能耗。文献[5]提出使用神经网络二值化训练方法,并用简单的加法操作替代复杂的乘累加操作。文献[6]利用权值共享技术,将权重值分类,数值相近的权重连接共用同一个值,以此达到缩减参数访存的目的。

(2)电路级。文献[7]等工作通过简化电路设计,来降低网络乘累加操作带来的能耗、性能开销。与此同时,研究人员发现通过调节处理器的供电电压,能够有效解决性能和能耗之间的平衡问题。动态电压调节(DVS)^[8]和动态电压频率调节(DVFS)^[9]是近年来流行的电压调节技术。然而,随着供电电压值逐渐降低到阈值电压附近,电路可靠性问题越来越严重。如图1所示,当部分运算单元(ALU)运行在足够低的电压下时,硬件运行时出现了错误,导致计算结果不精确。尽管如此,很多研究利用神经网络天然的容错能力,在降低电压的同时并没有极大地降低神经网络的输出质量。

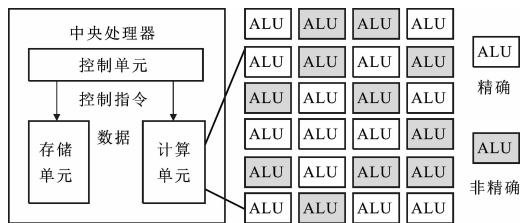


图1 CPU 结构图

3 神经元容错度量化方法

神经网络中不同神经元对错误的容忍能力是不同的,容错能力越大,说明错误发生在该神经元上时对节点计算结果的影响越小,反之亦然。为了量化神经元的容错能力,我们利用反向传播算法,以神经元输出值与代价函数的偏导值作为容错度的量化指标。值得注意的,从偏导值的计算公式可以看出,神经元容错能力与输入有关,如果更换一次输入就要重新计算每个神经元的容错能力将极大的浪费资源开销。所以提出基于神经元容错度的统计排序方法,修正神经元的容错度值,使得神经元的容错能力和输入无关,而和网络的本身结构相关。

3.1 神经元容错度计算

人工神经网络是由神经元按照一定结构相互连接而成的大规模复杂系统,具有很强的容错和自学能力。对于单一的神经元而言,节点输出为所有输入与权重的乘积和,为引入非线性因素,神经元输出需要通过激活函数。反向传播算法被广泛的应用于神经网络的学习过程,包含信号的前向传播和误差的反向传播^[10]。一个简单的神经网络自学传播关系如图2所示,其中 y_{ij} 表示第 i 层的第 j 个输入或输出, w_i 表示第 i 个输入所连接的到一下层的权重值, b_i 表示第 i 层相应神经元的偏置值, t 表示目标输出, E 表示代价函数。反向传播算法计算输出误差如何随输出值、输入值和权重而变化,从而将误差平均分配到网络的各层中,所以可以用来量化每个神经元对网络输出的错误贡献度。我们将独立神经元微小变化对神经网络输出质量的影响定义为容错度,用 δ 表示。根据反向传播算法,我们用神经元输出值和代价函数的偏导数作为容错度的评估指标^[4]。人工神经网络的代价函数种类丰富,本文以均方误差代价函数为例:

$$E = \frac{1}{2}(t - y_w)^2 \quad (1)$$

输出层神经元容错度 δ_{α} :

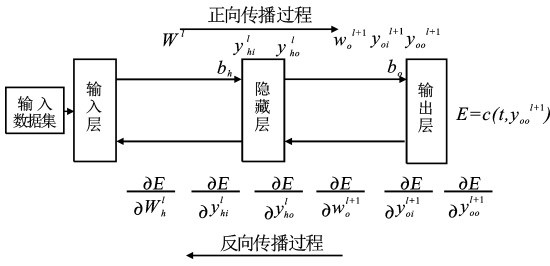


图2 神经网络自学传播关系

$$\delta_{oi} = \frac{\partial E}{\partial y_{oi}} \cdot \frac{\partial y_{oi}}{\partial y_{oi}'} = -(t - y_{oi}) \cdot f'(y_{oi}') \quad (2)$$

隐藏层神经元容错度 δ_{hi} :

$$\begin{aligned} \delta_{hi} &= \frac{\partial E}{\partial y_{oi}} \cdot \frac{\partial y_{oi}}{\partial y_{oi}'} \cdot \frac{\partial y_{oi}'}{\partial y_{hi}'} \cdot \frac{\partial y_{hi}'}{\partial y_{hi}} \\ &= \delta_{oi} \cdot w_{oi} \cdot f'(y_{hi}') \end{aligned} \quad (3)$$

输入层神经元容错度 δ_{xi} :

$$\begin{aligned} \delta_{xi} &= \frac{\partial E}{\partial y_{oi}} \cdot \frac{\partial y_{oi}}{\partial y_{oi}'} \cdot \frac{\partial y_{oi}'}{\partial y_{hi}'} \cdot \frac{\partial y_{hi}'}{\partial y_{hi}} \cdot \frac{\partial y_{hi}}{\partial y_{xi}} \\ &= \delta_{hi} \cdot w_{hi} \end{aligned} \quad (4)$$

如公式(2)~(4)所示,在训练阶段固定好所有的权重值 w 和偏置值 b 后,通过计算神经网络中每个神经元的容错度 δ_i ,并按照大小进行排序,即可得到神经元的容错度排序。

3.2 神经元容错度统计排序方法

实验发现,以隐藏层为例,将公式(3)中的 y_{hi} 继续展开得: $y_{hi} = w_{hi} \times x_i$,其中 x_i 表示输入。由此可以看出,神经元的输出值与网络的输入参数有关,基于不同输入所得到的排序结果会有差异,为了进一步提高排序结果的准确度,本文提出多次更换数据进行排序,然后基于统计算法选出最终排序结果。

ALGORITHM 1: 神经元容错度排序统计算法
输入: 完成预训练的神经网络 NN, 网络神经元数: Neuron-Num
测试数据集: TestData, 统计次数 epoch
输出: 排序结果

```

1: begin
2:   initialize; model = NN
3:   While iteration < epoch do
4:     random select data from TestData
5:      $\delta \leftarrow$  backpropagation (model, data)
6:      $\delta_{\text{sum}} \leftarrow$  sort ( $\delta$ )
7:   end while
8:   for  $k < \text{NeuronNum}$  do
9:     for  $j < \text{epoch}$ 
10:      neuron = max (statistic number of each neuron
                     in  $\delta_{\text{sum}}[j][k]$  and neuron not in sort)
11:      sort[ $k$ ] = neuron
12:   return sort

```

13: end

算法1描述了我们提出的统计算法。3~7行得到 epoch 次从小到大的排序结果,其中第4行从测试集中随机选取一组数据后,第5行将数据运行在神经网络模型上并使用反向传播算法得到每个神经元的容错度,第6行将排序后的结果存入 δ_{sum} 中。8~12行从小到大统计每个排序位置相应神经元出现的次数,当出现次数最多且没有之前排序中确定位置的神经元将作为本位置的最终神经元。由此,我们最终确定了隐藏层神经元的容错度排序。

为了证明排序的有效性,本实验多次更换排序输入并统计了32组隐藏层神经元容错度排序结果。从32组数据中任选一组排序中容错度最高的三个神经元(A、B、C)和容错度最低的三个神经元(D、E、F),并统计这六个神经元在排序结果中的分布情况,统计结果如图3所示,神经元在排序中的位置相对固定在特定范围,因此基于统计算法的神经元容错度排序结果的分布情况符合预期。最后,我们使用所有统计过程中容错度的均值来修正神经元的容错度。

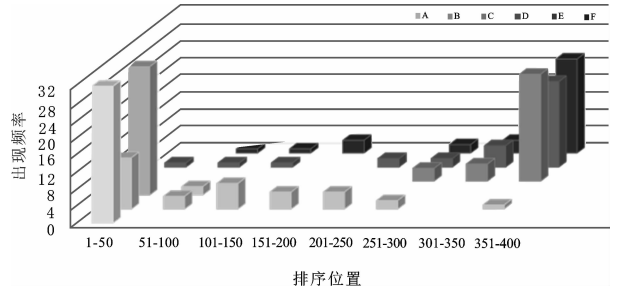


图3 神经元出现次数的分布

为了验证工艺偏差对网络输出质量的影响,本文模拟容错能力已知的神经元在不同质量电路上的运行情况,并统计神经网络的输出精度。神经元的容错能力表现为错误发生在该神经元上对网络输出精度的影响,容错能力越高,则错误的影响越小。结果如图4所示,错误发生在低容错能力的神经元上,且随着出错神经元个数的增加,网络的输出精度下降明显,相反,对于高容错能力的神经元而言网络精度的变化范围很小。因此可知,工艺偏差引发的错误发生在容错能力不同的神经元上对网络质量的影响不同。

4 神经元重要度分析及精度优化方法

传统的权重值裁剪方法不但会增加额外的数据记录稀疏矩阵的形式,并且会增加性能下降的风险^[1]。为了降低网络质量和性能的损失,尽可能压缩神经网络规模,本文采取节点裁剪方法,即通过裁剪

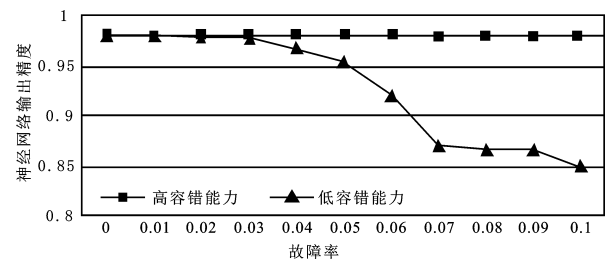


图4 相同错误下神经网络输出精度
网络中冗余的神经元达到降低神经网络资源需求的目的。

神经元裁剪方法的关键问题是如何在删除冗余神经元的同时满足精度需求. 裁剪的第一步是判断哪些神经元可以裁剪. 本文提出基于反向传播算法的神经元重要程度排序方法,用 ϵ 表示裁剪度,裁剪度越高说明相应神经元裁剪后对输出质量影响越低,反之亦然. 对于训练好的神经网络而言,采用梯度下降算法进行最优化求解时,网络参数收敛于全局极值点或局部极值点,而此时梯度的幅值也接近0. 即神经网络参数的梯度值越小,则该参数越接近极值点,该值越接近理想值. 以网络权重值更新为例:函数表示为 $y = w \times x + b$, w 的梯度为 $\nabla = x$;向梯度相反的方向移动 x ,如下: $x' \leftarrow x - \nabla \cdot \eta$,其中 η 为神经网络学习率;循环迭代上一步骤直到神经网络输入值 x 的变化使得 y 在两次迭代之间的差值足够小(接近0),则说明 y 接近局部最小值, w 接近理想值. 同理可推,神经元输出 y_i 与 E 的偏导值越小,则神经元输出 y_i 的值越接近理想值. 又因为 $y = w \times x + b$,则与 y_i 相连的上一层神经元的权重值与输入值的乘积越接近理想值,则可进一步认为与该神经元相连的上一层权重值越接近理想值. 节点裁剪方法中,当一个神经元被裁减,则与之相连接的所有权重连接都会消失. 如果裁剪接近理想值的神经元,则势必会裁剪掉接近理想值的权重矩阵,造成网络质量的下降. 所以针对节点裁剪方法,需要裁剪远离理想值的神经元. 分析可知,神经元的偏导数越大,裁剪度 ϵ 越高,神经元被裁剪后对输出精度的影响越小.

探索大型神经网络中哪些神经元冗余可以被裁剪的工作量非常巨大,且尽管重训练可以弥补裁剪引发的精度损失,但是输出层神经元之后没有权重值,因此训练算法更难降低这些神经元的误差,所以为了降低探索过程的工作量,提高网络裁剪的精度,本文将裁剪工作集中于神经网络的隐藏层. 隐藏层 ϵ 的计算表示如下:

$$\epsilon_i = \frac{\partial E}{\partial y_{hi}} \tag{5}$$

ϵ_i 值越大,则说明神经元对网络的重要程度越小,可裁剪程度越高,相反,则不可裁剪.

基于理论分析,我们得到神经元重要程度排序方法,与此同时为了最大化裁剪率使神经网络尺寸满足需求,我们引入重训练方法. 重训练过程有误差修复能力,基于此,我们提出循环裁剪法,来探寻最优裁剪率. 如图5所示,方法步骤如下:(1)通过计算隐藏层每个神经元 ϵ_i 值,得到网络的裁剪度排序;(2)根据裁剪度,移除最不重要的神经元;(3)有限次轻量级的重训练网络,补偿裁剪引起的输出质量的损失;(4)根据重训练的输出结果的精确度,调节裁剪率;(5)重复步骤(2)至(4)直到重训练过程不能将网络的质量恢复到阈值水平,则停止裁剪工作,输出网络低于阈值的前一次网络裁剪率.

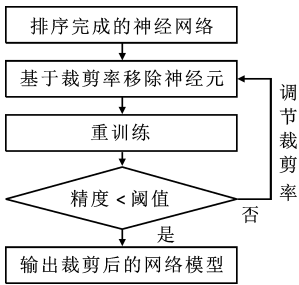


图5 循环裁剪流程图

5 实验结果

本节使用 MNIST、Fashion-MNIST、SVHN 和 CIFAR-10 四个数据集作为神经网络输入. MNIST 手写数字数据集包含60 000张训练数据和10 000张测试数据,每张图片的像素大小为 28×28 . Fashion-MNIST 数据集有10个类别共70 000张图片,其中60 000张训练图片,10 000张测试图片,每张图片的像素大小为 28×28 . 不同于 MNIST 的是,Fashion-MNIST 不再是抽象的数字符号,而是更加具象化的人类必需品——服装. SVHN 是对图像中阿拉伯数字进行识别的数据集,其中73 257张训练数字图片,26 032张测试数字图片.

CIFAR-10 数据集有10个类别共60 000张彩色图片,其中每类包含6 000张像素大小为 32×32 的RGB图片,50 000张图片用于训练,10 000张图片用于测试. 我们使用 PyTorch 作为神经网络软件级的运行环境. Sniper^[11] 模拟器被用于评估神经网络的运行性能. McPAT 工具被用于评估神经网络近似后的能耗开销.

5.1 神经元裁剪

MLP 是前馈神经网络的一种,由输入层、隐藏层和输出层组成,层与层之间通过全连接实现. MLP 隐藏层神经元数设为 $256 \rightarrow 256 \rightarrow 256$. 我们使用已有的 LeNet-5 结构作为 CNN 模型框架. MLP 与 CNN 网络详细信息见表 1. 将 MNIST 数据集运行在 MLP 上,所得神经网络精度损失为 1.9%. 我们分别测试基于 ϵ 排序、 δ 排序、随机排序和加入重训练过程的 ϵ 排序的节点裁剪方案. 结果如图 6(a) 所示,基于随机裁剪的神经网络输出精度浮动很大,因此该方法不适合应用到网络裁减中;基于 ϵ 排序的节点裁剪精度优于 δ 排序,证明上述基于 δ 的反序排序方法是合理的;在精度损失小于 1% 的情况下,MLP 的压缩率达到 50%;与此同时,重训练过程可以明显的补偿网路裁剪引起的精度损失问题,为进一步降低网络规模,节约硬件资源提供保障. 将 CIFAR-10 运行在 CNN 上,神经网络的精度损失为 49%. 实验结果如图 6(b) 所示,基于重训练的 ϵ 排序裁剪效果最优,在精度损失小于 1% 的情况下,神经网络的压缩率达到 10%;将精度损失提高到 5%,神经网络的压缩率达到 25%. CNN 的效果不如 MLP 的原因在于,CIFAR-10 数据集复杂程度远高于 MNIST 数据集,所以网络需要更多的参数维持识别精度.

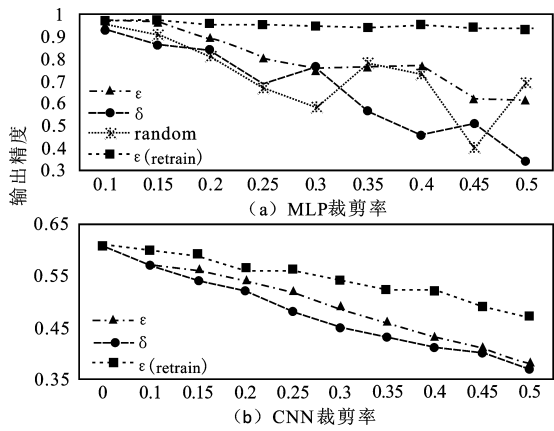


图 6 神经元裁剪精度

表 1 神经元裁剪实验数据表

Net works	Num of layers		Test Dataset	Accuracy rate
	CONV	FC		
MLP	0	5	MNIST	98.1%
LeNet-5	2	3	CIFAR-10	61%

5.2 近似计算

本节我们测试近似计算对神经网络输出质量的

影响. 为了最大限度的节约神经网络能耗开销,本实验选择电压调节的近似方法. 为了缓解电压调节技术引起的电路错误增多对神经网络输出质量的影响,我们根据神经元的容错能力,有选择的降低神经元运行时的电压. 值得注意的,神经网络中神经元个数非常多,如果每个神经元的电压都需要调节,则需要大量的电压调节器,这不但增加了硬件的面试开销,而且电压调节器的能耗开销也会影响神经网络最终的能耗.

为了降低电压调节器个数的影响,我们引入电压岛的概念,即相似容错能力的神经元分配同一个电压值,以此达到降低电压调节器的资源开销问题.

首先我们测试电压岛个数 k 对 MLP 的影响. 为了量化精度、能耗、性能对神经网络质量的影响,我们定义了网络质量的衡量指标 $M_NN = accuracy / (power * delay)$,其中 accuracy 表示神经网络的输出精度、power 表示神经网络运行过程中的能耗、delay 表示神经网络的运行时间. M_NN 取值越大,则说明网络质量越高. 在设定神经网络峰值能耗为 40 W 的情况下,不同 k 值下的变化情况如图 7 所示. 结果显示当 $k > 6$ 时,MNIST 数据集 M_NN 的输出质量大幅上升. 相比于其他孤岛的网络质量输出, $k = 16$ 是局部最优. 对于 Fashion-MNIAT、SVHN、CIFAR-10 而言,尽管存在一些局部最优,但是随着 k 值个数增多, M_NN 的趋势逐渐升高. 所以,为了降低电压调节的开销问题,在后续的实验,针对 MNIST 和 CIFAR-10 数据集而言, $k = 16$ 是较好的电压孤岛个数的取值;对于 SVHN 和 Fashion-MNIST 数据集而言, $k = 6$ 是较好的电压孤岛个数的取值.

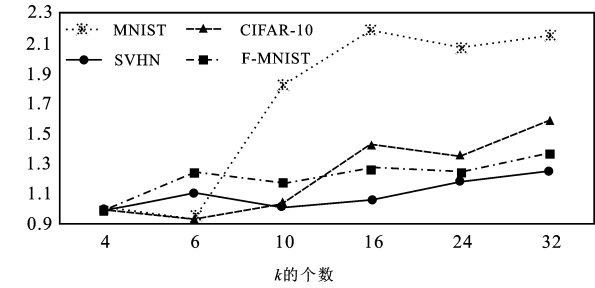


图 7 k 的个数对神经网络质量的影响

然后,我们以处理器的原始电压 2 V 和近阈值电压 0.5 V(粗粒度 coarse-grained, 方点)的整体网络的粗粒度电压调节作为参考. 普通的高性能低功耗的计算节点的峰值功耗是 55 W. 本实验中,我们将神经网络的峰值功耗分别设为 55 W、50 W、45 W

和 35 W,然后根据神经元的容错能力,等比例的降低神经元的供电电压(细粒度 fine-grained,圆点),使得神经网络的功耗满足上述限制.结果如图 8 所示,在 MLP 网络中,基于细粒度的电压调节的神经网络输出精度保持较高水平的同时,能耗和网络延迟也优于粗粒度的电压调节.特别地,数据集越简单、容错能力越高则近似后的神经网络质量越好.与 2 V 的原始供电电压下的神经网络能耗相比,对于 MNIST、Fashion-MNIST、SVHN 和 CIFAR-10 数据集而言,最好的细粒度的电压调节分别完成了 $1.35\times$ 、 $1.39\times$ 、 $1.27\times$ 、 $1.26\times$ 的节能效果.此外,当峰值功耗设为 45 W 时,对于 MNIST、Fashion-MNIST、SVHN 和 CIFAR-10 数据集而言,神经网络基于细粒度电压调节的输出精度是粗粒度级输出精度的 $63\times$ 、 $52\times$ 、 $1.1\times$ 和 $3\times$. 综上,细粒度的电压调节近似方法可以使神经网络的质量保持的更好.

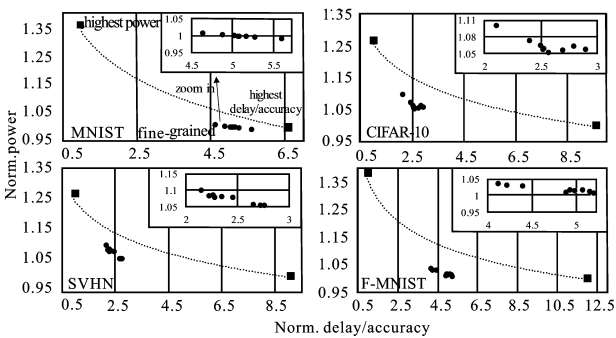


图 8 近似后的神经网络质量对比

6 结束语

本文根据神经元不同的容错能力,将神经元裁剪和近似计算技术相结合,以寻求最优的网络输出精度同时尽可能低的降低网络的能耗开销.本文主要优点在于(1)提出基于统计排序的神经元容错能力量化方法;(2)根据神经元的容错能力提出神经元重要程度排序算法;(3)在裁剪过程中引入重训练,提出循环裁剪法;(4)在神经网络运行过程中使用近似计算技术进一步降低功耗开销.最后,通过两个实验,证明了该技术的有效性.

参考文献:

[1] YU J, LUKEFAHR A, PALFRAMAN D, et al. Scalpel: customizing DNN pruning to the underlying hardware parallelism[J]. AcmSigarch Computer Architecture News, 2017, 45(2):548-560.

[2] BABAEIZADEH M, SMARAGDIS P, CAMPBELL R H. NoiseOut: a simple way to prune neural networks [J]. arXiv. org > cs > arXiv: 1611.06211v1, 2016.

[3] YANG T J, CHEN Y H, SZE V. Designing energy-efficient convolutional neural networks using energy-aware pruning[C]// Computer Vision & Pattern Recognition. Honolulu, HI, USA, IEEE, 2017.

[4] ZHANG Q, WANG T, TIAN Y, et al. Approx-ANN: an approximate computing framework for artificial neural network[C]// Design, Automation & Test in Europe Conference & Exhibition. Grenoble, France, EDA Consortium, 2015.

[5] COURBARIAUX M, BENGIO Y, DAVID J P. BinaryConnect: training deep neural networks with binary weights during propagations[C]// International Conference on Neural Information Processing Systems. Istanbul, Turkey, 2015.

[6] HAN S, MAO H, DALLY W J. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding[J]. Fiber, 2015, 56(4):3-7.

[7] SARWAR S S, VENKATARAMANI S, RAGHUNATHAN A, et al. Multiplier-less artificial neurons exploiting error resiliency for energy-efficient neural computing[C]// 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE). Dresden, Germany, IEEE, 2016.

[8] 邢静宇, 张立臣. 动态电压调整多处理器实时系统任务调度[J]. 微电子学与计算机, 2006, 23(2):55-57.

[9] GUERREIRO J, ILIC A, ROMA N, et al. GPGPU power modeling for multi-domain voltage-frequency scaling[C]// IEEE International Symposium on High Performance Computer Architecture. Vienna, Austria, 2018.

[10] 朱新召, 胡哲琨, 周莉, 等. 基于多核处理器的多层感知神经网络设计和实现[J]. 微电子学与计算机, 2014, 31(11):27-31.

[11] CARLSON T E, HEIRMAN W, EECKHOUT L. Sniper: exploring the level of abstraction for scalable and accurate parallel multi-core simulation[C]// High Performance Computing, Networking, Storage & Analysis. Washington, USA, IEEE, 2011.

作者简介:

王旭男, (1993-), 硕士研究生. 研究方向为计算机系统结构、容错计算.

试验均处于白天,并没有考虑夜间低照度下对物体的跟踪算法,未来无人机可以结合红外探测装置对夜间目标跟踪做进一步研究。

参考文献:

- [1] COŞKUN M, ÜNAL S. Implementation of tracking of a moving object based on camshift approach with a UAV [J]. *Procedia Technology*, 2016(22):556-561.
- [2] 袁国武. 智能视频监控中的运动目标检测和跟踪算法研究[D]. 昆明:云南大学,2012.
- [3] 屈晶晶,辛云宏.连续帧间差分与背景差分相融合的运动目标检测方法[J]. *光子学报*,2014,43(7):219-226.
- [4] 贺若彬,武德安,吴磊,等.稀疏子空间的粒子滤波法在目标跟踪中的应用[J]. *计算机工程与设计*,2016,37(11):3080-3085.
- [5] 李成美,白宏阳,郭宏伟,等.一种改进光流法的运动目标检测及跟踪算法[J]. *仪器仪表学报*,2018,39(5):249-256.
- [6] 赵欣,陈峰,吴立知.一种改进的 meanshift 运动目标跟踪算法[J]. *通信技术*,2011,44(11):76-79.
- [7] LI L, LUO Y. Improved video moving target tracking based on camshift[J]. *American Journal of Computational Mathematics*, 2016, 6(4):357-364.
- [8] XIU C, WEI S, WAN R, et al. Camshift tracking method based on target decomposition[J]. *Mathematical Problems in Engineering*, 2015(1):1-20.
- [9] GUO L, LI L, ZHAO Y, et al. Pedestrian tracking based on camshift with kalman prediction for autonomous vehicles[J]. *International Journal of Advanced Robotic Systems*, 2016(13):1-5.
- [10] 翟卫欣,程承旗.基于 Kalman 滤波的 Camshift 运动跟踪算法[J]. *北京大学学报(自然科学版)*, 2015, 51(5):799-804.
- [11] BAN L Y, HAN Y H, WU Y H. Target tracking based on improved camshift and kalman filter[J]. *Ad-*

vanced Materials Research, 2014 (989/994): 3587-3590.

- [12] 宋晓琳,王文涛,张伟伟.基于 LBP 纹理和改进 Camshift 算子的车辆检测与跟踪[J]. *湖南大学学报(自然科学版)*, 2013, 40(8):52-57.
- [13] WU X, LI L, LAI J, et al. A framework of face tracking with classification using CAM Shift-C and LBP[C]// *Fifth International Conference on Image and Graphics*. China, Xi' an, IEEE Computer Society, 2009:217-222.
- [14] YUAN G W, GAO Y, XU D. A moving objects tracking method based on a combination of local binary pattern texture and hue[M]. *Springer Berlin Heidelberg, Electrical Engineering and Control*, 2011: 3964-3968.
- [15] 初红霞,谢忠玉,王君祥,等.空间相关背景加权直方图均值漂移目标跟踪[J]. *控制与决策*, 2014, 29(3): 528-532.
- [16] 刘亚伟,李小民,陈为元.基于改进 CamShift 融合 Kalman 滤波的无人机目标跟踪研究[J]. *光电与控制*, 2017,24(8):33-37.
- [17] 修春波,魏世安,万蓉凤.二维联合特征模型的自适应均值漂移目标跟踪[J]. *光电子·激光*, 2015, 26(2): 342-351.
- [18] 覃跃虎,支琤,徐奕.基于三维直方图的改进 Camshift 目标跟踪算法[J]. *现代电子技术*, 2014, 37(2):29-33.

作者简介:

沈思源 男,(1993-),硕士研究生.研究方向为图像处理、模式识别.

李震霄 男,(1995-),硕士研究生.研究方向为图像处理.

孙伟(通讯作者) 男,(1963-),教授,博士生导师.研究方向为图像处理、智能控制、模式识别.

E-mail:neuqssy@163.com

(上接第 75 页)

王晶(通信作者) 女,(1982-),博士,副研究员.研究方向为计算机系统结构、高性能计算、容错计算、智能芯片设计.

E-mail:jwang@cnu.edu.cn.

张伟功 男,(1967-),博士,研究员.研究方向为高可靠计算机体系结构及 SoC 设计、高速高可靠总线技术、计算机容错技术.