**神经网络量化白皮书**

### 马库斯·内格尔∗

高通AI研究†

[markusn@qti.qualcomm.com](mailto:markusn@qti.qualcomm.com)

马里奥·福尔纳拉基斯∗高通人工智能研究†[mfournar@qti.qualcomm.com](mailto:mfournar@qti.qualcomm.com)

### 拉纳·阿里·阿姆贾德

高通AI研究†

[ramjad@qti.qualcomm.com](mailto:ramjad@qti.qualcomm.com)

叶丽赛·邦达连科高通AI研究†[ybodaren@qti.qualcomm.com](mailto:ybodaren@qti.qualcomm.com)

Mart van Baalen高通AI研究†[mart@qti.qualcomm.com](mailto:mart@qti.qualcomm.com)

蒂杰曼·布兰克沃特高通AI研究†[tijmen@qti.qualcomm.com](mailto:tijmen@qti.qualcomm.com)

# 摘要

虽然神经网络在许多应用领域取得了进展，但它们往往具有很高的计算成本。如果我们想要将现代网络整合到对功率和计算要求严格的边缘设备中，降低神经网络推理的功率和延迟是关键。神经网络量化是实现这些节省的最有效方法之一，但它所产生的额外噪声会导致精度下降。

在这份白皮书中，我们介绍了最先进的算法，用于在保持低比特权重和激活的同时减轻量化噪声对网络性能的影响。我们首先介绍了硬件驱动的量化，然后考虑了两类主要算法：训练后量化（PTQ）和量化感知训练（QAT）。PTQ不需要重新训练或标记数据，因此是一种轻量级的按钮量化方法。在大多数情况下，PTQ足以实现接近浮点精度的8位量化。QAT需要微调和访问标记的训练数据，但能够实现较低的比特量化，并具有竞争性的结果。对于这两种解决方案，我们都提供基于现有文献和广泛实验的测试管道，从而为常见的深度学习模型和任务提供最先进的性能。

# 导言

随着深度学习作为向电子设备注入智能的通用工具的日益普及，对小型、低延迟和节能的神经网络解决方案的需求也越来越大。如今，神经网络可以在许多电子设备和服务中找到，从智能手机、智能眼镜和家用电器，到无人机、机器人和自动驾驶汽车。这些设备通常受到严格的神经网络执行时间限制或对长时间性能的严格功率要求。

减少神经网络计算时间和能量消耗的最有影响的方法之一是量化。在神经网络量化中，权值和激活张量的存储精度低于它们通常训练的16或32位精度。当

∗同等贡献。 †高通人工智能研究是高通技术公司的一项举措。

从32位移动到8位，存储张量的内存开销减少了4倍，而矩阵乘法的计算成本减少了16倍。神经网络已被证明对量化具有鲁棒性，这意味着它们可以被量化到较低的比特宽度，而对网络精度的影响相对较小。此外，神经网络量化通常可以与其他常用的神经网络优化方法一起应用，如神经网络结构搜索、压缩和剪枝。对于任何实际的深度学习用例来说，这都是模型效率管道中必不可少的一步。然而，神经网络量化并不是免费的。低比特宽度量化会给网络带来噪声，从而导致精度下降。虽然一些网络对这种噪声具有鲁棒性，但其他网络需要额外的工作来利用量化的好处。

在这篇白皮书中，我们介绍了神经网络量化的最新技术。我们从量化的介绍开始，讨论硬件和实际考虑。然后，我们考虑了两种不同的量化神经网络模式：训练后量化(PTQ)和量化感知训练(QAT)。PTQ方法，在第#节中讨论[3](#_bookmark25)，采用训练好的网络，用很少或没有数据对其进行量化，只需要极少的超参数调整，不需要端到端的训练。这使得它们成为一种按下按钮的方法，以低工程努力和计算成本量化神经网络。与此相反，QAT，在第#节中讨论[4](#_bookmark57)，依赖于在训练管道中用模拟量化对神经网络进行再训练。虽然这需要在训练和潜在的超参数调谐方面付出更多的努力，但与用于低比特量化的PTQ相比，它通常进一步缩小了与全精度精度的差距。对于这两种机制，我们都基于现有文献和广泛的实验引入了标准管道，这些管道导致了通用计算机视觉和自然语言处理模型的最先进性能。我们还提出了一个调试工作流，以识别和解决在量化新模型时常见的问题。

# 量化基础

在这一节中，我们介绍了神经网络量化的基本原理以及量化网络运行的定点加速器。我们首先从硬件的动因开始，然后介绍标准量化方案及其性质。我们稍后讨论与现代神经网络中常见的层相关的实际考虑因素及其对定点加速器的影响。

### 硬件背景

在深入研究技术细节之前，我们首先探讨了量化的硬件背景，以及它如何在设备上实现高效的推理。图[1](#_bookmark4) 提供了如何在神经网络（NN）加速器中计算矩阵——向量乘法（y=Wx+b）的示意性概述。这是神经网络中更大的矩阵——矩阵乘法和卷积的组成部分。这样的硬件块旨在通过并行执行尽可能多的计算来提高NN推理的效率。该NN加速器的两个基本部件是处理元件C*n,m*和累加器A*n*。我们的玩具例子[1](#_bookmark4) 有16个方形网格状的处理元件和4个蓄能器。计算从用偏置值b*n*加载累加器开始。然后，我们将权重值W*n,m*和输入值x*m*加载到数组中，并在单个循环中计算它们在各自处理元素C*n,m*=W*n,m*x*m*中的乘积。然后将其结果添加到累加器中：

A*n* = b*n* + *Cn,m* (1)

*m*

上述操作也称为乘——累加（MAC）。对于较大的矩阵——向量乘法，此步骤重复多次。一旦所有循环完成，累加器中的值就会被移回内存，以便在下一个神经网络层中使用。神经网络通常使用FP32权值和激活进行训练。如果我们要在FP32中执行推理，处理元件和累加器必须支持浮点逻辑，并且我们需要将32位数据从存储器传输到处理单元。MAC操作和数据传输消耗了神经网络推理过程中消耗的大部分能量。因此，通过对这些量使用较低比特定点或量化表示，可以获得显著的好处。低比特定点表示，例如INT8，不仅减少了数据量传输，而且减少了MAC操作的大小和能耗([霍洛维茨，2014年](#_bookmark80)）。这是

因为数字算法的成本通常随所用比特数线性地扩展到二次型，并且因为定点加法比浮点加法更有效([霍洛维茨，2014年](#_bookmark80)).

输入值

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | x! | x# | x$ | x% |  |
| W!,! W!,# W!,$ W!,% | #!,! | #!,# | #!,$ | #!,% | $! |
| W#,! W#,# W#,$ W#,% | ##,! | ##,# | ##,$ | ##,% | $# |
| W$,! W$,# W$,$ W$,% | #$,! | #$,# | #$,$ | #$,% | $$ |
| W%,! W%,# W%,$ W%,% | #%,! | #%,# | #%,$ | #%,% | $% |

图1：神经网络加速器硬件中矩阵乘法逻辑示意图概述。

为了从浮点到高效的定点操作，我们需要一个将浮点向量转换为整数的方案。浮点向量x可以近似表示为标量乘以整数值的向量：

x = s**x** · x国际 ≈ x

(2)

其中s**x**是浮点比例因子，x国际是整数向量，例如INT8。我们将这个量化版本的矢量表示为x。通过量化权重和激活，我们可以编写累积方程的量化版本：

*A*ˆ*n* = **b**.*n* + L **W** *n,m* **x**.*m*

=bn+Ls**w**W国际

*sxxint*

*m*

=bn+s**w**s**x**LW国际

国际

*m*

(3)

*m*

注意，我们对权重s**w**和激活s**x**使用了单独的比例因子。这提供了灵活性并减少了量化误差（更多见第#节）[2.2](#_bookmark6)）。由于每个比例因子都应用于整个张量，因此该方案允许我们将比例因子因子从方程中的和中因子([3](#_bookmark5)）并以定点格式执行MAC操作。我们暂时有意忽略偏置量化，因为偏置通常存储在较高的位宽（32位）中，其比例因子取决于权重和激活因子（[雅各布等人，2018](#_bookmark83)).

图[2](#_bookmark7) 显示了当我们引入量化时，神经网络加速器是如何变化的。在我们的例子中，我们使用INT8算法，但为了讨论，这可以是任何量化格式。重要的是，要保持较高的位宽的累加器，典型的32位宽。否则，由于计算期间累积更多产品，我们面临因溢利而产生亏损的风险。

存储在32位累加器中的激活需要写入内存，然后才能被下一层使用。为了减少数据传输和下一层操作的复杂性，这些激活被量化回INT8。这需要一个重新量化步骤，如图所示[2](#_bookmark7).

### 均匀仿射量化

在这一部分中，我们定义了本文将使用的量化方案。这种方法被称为均匀量化，是最常用的量化方法，因为它允许有效地实现定点算法。

输入值

!" ⋅

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| W!,! W!,# W!,$ W!,% | #!,! | #!,# | #!,$ | #!,% | $! |
| W#,! W#,# W#,$ W#,%  !! ⋅ | ##,! | ##,# | ##,$ | ##,% | $# |
| W$,! W$,# W$,$ W$,% | #$,! | #$,# | #$,$ | #$,% | $$ |
| W%,! W%,# W%,$ W%,% | #%,! | #%,# | #%,$ | #%,% | $% |
| INT8 |  |  |  |  | INT32 |

图2：用于量化推理的神经网络加速器中的矩阵乘法逻辑示意图。

均匀仿射量化又称非对称量化，由三个量化参数定义：尺度因子s、零点z和位宽b。比例因子和零点用于将浮点值映射到整数网格，整数网格的大小取决于位宽。比例因子通常表示为浮点数，并指定量化器的步长。零点是一个整数，它确保真实的零点被量化而不会出错。这对于确保像零填充或ReLU这样的常见操作不会引起量化误差非常重要。

一旦定义了三个量化参数，我们就可以继续进行量化操作。从实值向量x开始，我们首先将其映射到无符号整数网格{0,.,2*b*−1}：

x国际=夹具( x 1+z；0.2*b*-1)，(4)其中l·l是圆到最接近算子，夹紧定义为：

夹具(x;a,c)=

*x, a* ≤ *x* ≤ *c,*

*c, x > c.*

(5)

为了近似真实值的输入x，我们执行去量化步骤：

x ≈ x = s (x国际 − z)

(6)

结合上述两个步骤，我们可以提供量化函数的一般定义，

*q*(·), as:

x=q(x;s,z,b)=s夹具 **x** + *z*; 0*,* 2*b* 1 *z ,* (7)

*s*

通过去量化步骤，我们还可以定义量化网格极限（qmin，q最大），其中qmin=sz，q最大=s（2*b*1Z）。超出此范围的任何x值都将被裁剪到其极限，从而产生裁剪错误。如果要减小剪切误差，可以通过增大比例因子来扩大量化范围。但是，增加比例因子会导致

详细介绍了如何选择量化参数，实现裁剪误差和舍入误差之间的正确权衡。

### 对称均匀量化

对称量化是一般非对称情形的简化版本。对称量化器将零点限制为0。这减少了在等式([3](#_bookmark5)）。但是缺少偏移限制了整数域和浮点域之间的映射。因此，有签名或无签名的选择

整数网格重要：

**x = s x**国际

x国际=夹具

( **x**

1; 0*,* 2*b* − 1)

(8a)

对于无符号整数 (8b)

x国际=夹具 **x** ; 2*b*−1*,* 2*b*−1 1 对于有符号整数 (8c)

*s*

无符号对称量化非常适合于单尾分布，如ReLU激活（见图[3](#_bookmark11)）。另一方面，对于大约为零的分布，可以选择有符号对称量化。

## 对称有符号 对称无符号

*s* **x**int8 *s* **x**uint8

0 斧头

## 不对称的

*s(x*uint8 *- z)*

图3：位宽为8的不同均匀量化网格的可视化说明。s是缩放因子，z是零点。浮点网格为黑色，整数量化网格为蓝色。

### 二倍功率量化器

二次方量子化是对称量子化的一种特殊情况，其中比例因子被限制为二次方，s=2−k。这种选择可以带来硬件效率，因为使用s进行缩放对应于简单的位移位。然而，尺度因子的有限表达会使舍入误差和剪裁误差之间的权衡变得复杂。

### 量化粒度

到目前为止，我们已经定义了每个张量的一组量化参数（量化器），一个用于权值，一个用于激活，如等式（[3](#_bookmark5)）。这被称为每张量量化。我们还可以为张量的各个段(例如权重张量的输出通道)定义单独的量化器，从而增加量化粒度。在神经网络量化中，由于其硬件实现更简单，每张量量化是最常见的粒度选择：方程中的所有累加器（[3](#_bookmark5)）使用相同的比例因子s**w**s**x**。但是，我们可以使用更细的粒度来进一步提高性能。例如，对于权值张量，我们可以为每个输出通道指定不同的量化器。这就是所谓的每通道量化，其含义将在本节中更详细地讨论[2.4.2](#_bookmark24).

其他工作超越了每个信道量化参数，并为每个组权重或激活应用单独的量化器（[鲁哈尼等人，2020](#_bookmark92); [斯托克等人，2019年](#_bookmark96); [纳西门托等人，2019年](#_bookmark90)）。增加组的粒度通常会以增加一些额外开销为代价来提高准确性。开销与处理具有不同比例因子的值和的累加器相关联。大多数现有的定点加速器目前不支持这种逻辑，因此，我们在这项工作中不会考虑它们。然而，随着这方面研究的增长，未来可以预期对这些方法有更多的硬件支持。

### 量化模拟

为了测试神经网络在量化设备上的运行效果，我们经常在用于训练神经网络的同一通用硬件上模拟量化行为。这叫做

输出

int8

输出

int32

int32

int32

int8

int8

输入

1. 用于定点操作的量化设备上推断的图。

输入

1. 使用浮点运算的模拟量化。

图4：卷积层的量化前向传递示意图：a）实际设备上量化推理的计算图。b)通用浮点硬件量化推理仿真。

量化模拟。我们的目标是使用浮点硬件近似定点操作。与在实际的量化硬件上运行实验或使用量化内核相比，这种模拟要容易得多。它们允许用户有效地测试各种量化选项，并且它能够为量化感知训练提供GPU加速，如第1节所述[4](#_bookmark57)在本节中，我们首先解释了此仿真过程的基本原理，然后讨论了有助于减少仿真与实际设备性能之间差异的技术。

之前我们看到了矩阵——向量乘法是如何在专用定点硬件中计算的。图中[4a](#_bookmark14)，我们将这个过程推广到一个卷积层，但我们也包括了一个激活函数，使其更加真实。在设备上推理期间，所有输入（偏置、权重和输入激活）都是以定点格式输入到硬件的。然而，当我们使用通用深度学习框架和通用硬件模拟量化时，这些数量处于浮点。这就是为什么我们在计算图中引入量化器块来诱导量化效应。

图[4b](#_bookmark14) 展示了如何在深度学习框架中对同一卷积层进行建模。在权值和卷积之间加入量化器块来模拟权值量化，在激活函数之后加入量化器块来模拟激活量化。由于偏差存储的精度较高，因此通常不会对其进行量化。在节中[2.3.2](#_bookmark20)我们将更详细地讨论何时将量化器定位在非线性之后是合适的。量化器块实现了方程的量化功能([7](#_bookmark9)并且每个量化器由一组量化参数(比例因子、零点、位宽)定义。量化器的输入和输出都是浮点格式，但输出位于量化网格上。

### 批量归一化折叠

批量规范化([Ioffe&Szegedy，2015年](#_bookmark82)）是现代卷积网络的标准组件。批量规范化在缩放和添加偏移之前对线性层的输出进行规范化（参见等式[9](#_bookmark18)）。对于设备上的推断，这些操作在称为批量规范化折叠([Krishnamoorthi，2018年](#_bookmark85); [雅各布等人，2018](#_bookmark83)）。这将从网络中完全删除批规范化操作，因为计算被吸收到相邻的线性层中。除了减少额外的缩放和偏移的计算开销外，这还防止了额外的数据移动和层输出的量化。更多

在形式上，在推理过程中，批量规范化被定义为输出x的仿射映射：

(  **x** − *µ* }

其中µ和σ是训练期间作为批统计量上的指数移动平均计算的平均值和方差，γ和β是每个通道学习的仿射超参数。如果批量规范化是在线性层y=BatchNorm(Wx)之后应用的，我们可以重写术语，使批量规范化操作与线性层本身融合。假设权重矩阵W∈R*n×m*，我们对每个输出y*k*应用批量规范化，对于k={1,.,n}：

y*k*=批次标准(W*k,:*x)

=γ K ***σ***2 + *E*

+ ***β****k*

***γ****k* **W***k,*:

= ***σ***2 + *E* **x** +

***γ µ β K−σ***2 ***+ E***

其中：

= **W** *k,*: **x** + **b** *k* (10)

***γ****k* **W***k,*:

**W***k,:* **=** 2 *,*

*k*

**b  *γ****k* ***µ****k*

*k* = ***β****k* 2 *.*

*k*

### 激活函数融合

在我们的朴素量子加速器中[2.1](#_bookmark2)我们看到，在计算矩阵乘法或卷积输出值之后，激活的再量化发生了。然而，在实际中，我们往往在线性运算之后直接出现非线性。将线性层的激活写入内存，然后将它们加载回计算核心以应用非线性，这是一种浪费。为此，许多硬件解决方案都带有一个硬件单元，该单元在重新量化步骤之前应用非线性。如果是这样，我们只需要模拟非线性之后发生的再量化。例如，ReLU非线性可以通过requantization块很容易地建模，因为您可以将激活量化的最小可表示值设置为0。

其他更复杂的激活功能，如sigmoid或Swish([Ramachandran等人，2017年](#_bookmark91)），需要更多的专门支持。如果这个支持不具备，我们需要在图中非线性前后增加一个量化步骤。这会对量化模型的精度产生很大的影响。虽然较新的激活如Swish功能提供了浮点精度的改进，但是这些在量化之后可能消失，或者在定点硬件上部署可能效率较低。

### 其他层和量子化

神经网络中使用了许多其他类型的层。这些模型的建模方式在很大程度上取决于具体的硬件实现。有时，模拟量化和目标性能之间的不匹配归结为没有正确量化层。在这里，我们提供一些指导，说明如何模拟几个常用层的量化：

不需要最大池激活量化，因为输入和输出值位于同一量化网格上。

平均池化整数的平均值不一定是整数。因此，在平均池化之后需要一个量化步骤。然而，我们对输入和输出使用相同的量化器，因为量化范围没有显著变化。

元素相加尽管性质简单，但这种操作很难精确模拟。在加法过程中，两个输入的量化范围必须完全匹配。如果这些范围

如果不匹配，则需要额外的注意，以使附加工作按预期进行。对此没有单一的可接受的解决方案，但是添加一个重量化步骤可以粗略地模拟添加的噪声。另一种方法是通过绑定输入的量化网格来优化网络。这将防止再量化步骤，但可能需要微调。

连接两个连接的分支通常不共享相同的量化参数。这意味着它们的量化网格可能不重叠，使得重新量化步骤是必要的。与元素相加一样，可以优化网络，使其具有连接分支的共享量化参数。

### 实际考虑

在对多层神经网络进行量化时，我们面临着大量的量化选择，包括量化方案、粒度和比特宽度。在本节中，我们将探讨一些有助于减少搜索空间的实际考虑因素。

注意，在本白皮书中，我们只考虑了均匀位宽。这意味着为权重或激活选择的位宽在所有层中保持恒定。同质位宽在硬件上得到了更普遍的支持，但最近的一些工作也探索了异质位宽或混合精度的实现（[范巴伦等人，2020](#_bookmark99); [董等，2019](#_bookmark76); [Uhlich等人，2020](#_bookmark97)).

### 对称与非对称量化

对于每一次权值和激活量化，我们都要选择一种量化方案。一方面，非对称量化由于存在额外的偏移参数，因此更具表现力，但另一方面也存在可能的计算开销。要了解为什么会这样，请考虑当不对称权重W--=s**w**W国际-z**w**)乘以不对称激活时会发生什么

= s**x**(x国际 − z**x**)

*=s***w***s***x***W*国际*x*国际*-s***w***z***w***s***x***x*国际*-s***w***s***x***z***x***W*国际*+s***w***z***w***s***x***z***x***。* (13)

第一项是如果两个运算都是对称格式的话我们会得到的。第三项和第四项仅取决于比例、偏移和权重值，这些值是预先已知的。因此，这两个项可以预先计算并添加到层的偏置项中，几乎不需要任何成本。然而，第二项取决于输入数据x。这意味着，对于每一批数据，我们需要在推理过程中计算一个额外的项。这会导致延迟和功率方面的重大开销，因为这相当于增加了一个额外的信道。

因此，使用不对称激活量化和对称权量化是一种常见的方法，避免了附加的数据依赖项。

### 每张量和每通道量化

在节中[2.2.3](#_bookmark12)，我们讨论了不同级别的量化粒度。权值和激活的每张量量化已经成为标准一段时间了，因为它被所有定点加速器所支持。然而，权重的每信道量化可以提高精度，尤其是当权重的分布在信道之间显著变化时。回顾方程中的量化MAC操作（[3](#_bookmark5)），我们可以看到，通过应用单独的每通道权重比例因子，可以在加速器中实现每通道权重量化，而不需要重新缩放。激活的每通道量化要难得多，因为我们不能将比例因子从求和中因子出来，因此需要为每个输入通道重新缩放累加器。虽然每个信道的权值量化越来越成为一种普遍做法，但并非所有商用硬件都支持它。因此，检查目标设备中是否有可能这样做是很重要的。

# 训练后量化

后训练量化（PTQ）算法取一个预训练的FP32网络，直接转换成固定点网络，不需要原有的训练流水线。这些方法可以

|  |  |  |
| --- | --- | --- |
| 型号（FP32精度） | ResNet18 (69.68) | 移动电视2（71.72） |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 位宽 | W8 | W6 | W4 | W8 | W6 | W4 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 最小值-最大值 | **69.57** | 63.90 | 0.12 | **71.16** | 64.48 | 0.59 |
| MSE | 69.45 | **64.64** | **18.82** | 71.15 | **65.43** | **13.77** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 最小值-最大值（每个通道） | 69.60 | 69.08 | 44.49 | 71.21 | 68.52 | 18.40 |
| MSE（每通道） | **69.66** | **69.24** | **54.67** | **71.46** | **68.89** | **27.17** |

表1：在FP32中保持激活的同时，不同方法设置（对称均匀）权量化器范围的烧蚀研究。5次运行以上的平均ImageNet验证准确率(%)。

不需要数据，或者可能需要一个小型校准装置，而这通常是现成的。此外，几乎没有超参数调整使得它们可以通过单个API调用作为黑箱方法以计算效率的方式量化预训练的神经网络。这使得神经网络设计者不必成为量化专家，从而允许神经网络量化的更广泛的应用。

PTQ过程中的一个基本步骤是为每个量化器找到良好的量化范围。我们在第#节中简要讨论了[2.2](#_bookmark6) 量化范围的选择如何影响量化误差。在这一部分，我们从讨论各种在实践中常用的寻找好的量化参数的方法开始。然后，我们探讨了PTQ期间观察到的常见问题，并介绍了克服这些问题的最成功的技术。利用这些技术，我们提出了一个标准的训练后量化管道，我们发现它在大多数常见的场景中工作得最好，最后我们引入了一组调试步骤来提高量化模型的性能。

### 量化范围设置

量化范围设置是指确定量化网格的剪裁阈值qmin和q最大的方法(见等式[7](#_bookmark9)）。在范围设置中，关键的权衡是在裁剪误差和舍入误差之间，在第#节中描述[2.2](#_bookmark6)，以及它们对每个配置的量化器的最终任务损失的影响。这里描述的每一种方法提供了两个量之间的不同的权衡。这些方法通常优化局部成本函数而不是任务损失。这是因为在PTQ中，我们的目标是计算速度快的方法，而不需要端到端的训练。权重通常可以量化，而不需要任何校准数据。然而，确定激活量化的参数通常需要几批校准数据。

最小—最大为了覆盖张量的整个动态范围，我们可以定义如下量化参数

qmin=min V，(14)

q最大=最大V，(15)

其中V表示要量化的张量。这不会导致剪切错误。然而，这种方法对异常值很敏感，因为强异常值可能会导致过度舍入误差。

均方误差（MSE）一种缓解大异常值问题的方法是使用基于MSE的范围设置。在这种范围设置方法中，我们发现qmin和q最大最小化了原始张量和量化张量之间的MSE：

最小参数IV-V（q *, q* )I *,* (16)

其中V（qmin，q最大）表示V的量化版本，*F*是Frobenius范数。优化问题通常采用网格搜索、黄金分割法或闭式解析近似求解。[班纳等人，2019年](#_bookmark71)）。此范围设置的几种变体

文献中也有方法，但它们在目标函数和最优化方面都非常相似。

|  |  |  |
| --- | --- | --- |
| 型号（FP32精度） | ResNet18 (69.68) | 移动电视2（71.72） |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 位宽 | A8 | A6 | A4 | A8 | A6 | A4 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 最小值-最大值 | 69.60 | 68.19 | 18.82 | 70.96 | 64.58 | 0.53 |
| MSE | 69.59 | 67.84 | 31.40 | 71.35 | 67.55 | 13.57 |
| MSE + Xent | **69.60** | **68.91** | **59.07** | **71.36** | **68.85** | **30.94** |
| BN (*α* = 6) | 69.54 | 68.73 | 23.83 | 71.32 | 65.20 | 0.66 |

表2：在保持FP32中的权值的同时，不同方法设置（非对称均匀）激活量化器范围的烧蚀研究。5次运行以上的平均ImageNet验证准确率(%)。

交叉熵对于某些层，被量化的张量中的所有值可能不是同等重要。其中一种情况是对分类网络最后一层的logits进行量化，在这种情况下，重要的是在量化后保持最大值的顺序。MSE可能不是一个合适的度量，因为它对张量中的所有值都进行了同等的加权，而不管它们的顺序如何。对于数量较多的类，我们通常有大量对预测精度不重要的小或负对数，也很少有重要的大值。在这种情况下，MSE会对少数较大的重要对数产生较大的量化误差，同时试图减小较小的对数的量化误差。在这种特定情况下，最小化以下交叉熵损失函数是有益的

最小参数

qmin、qmax

H (ψ (v) , ψ (vˆ(qmin, q最大))) (17)

其中H(·，·)表示交叉熵函数，ψ是软最大函数，v是对数向量。基于BN的量程设置激活量化器的量程设置通常需要一些校准数据。如果一层具有批规范化激活，则激活的每通道平均值和标准差分别等于已学习的批规范化移位和缩放参数。然后，这些可以用于找到激活量化器的合适参数，如下所示（[Nagel等人，2019](#_bookmark88)):

qmin=min(β-α γ) (18)

q最大=最大值(β+α γ) (19)

其中β和γ是每个通道学习的移位和尺度参数的矢量，并且α>0。[Nagel等人（2019年](#_bookmark88)）使用α=6，这样只裁剪大的异常值。

**表中的比较**[1](#_bookmark26) 比较了权值量化的范围设置方法。对于高位宽，MSE和min-max方法大多是对的。然而，在较低的位宽下，MSE方法的性能明显优于最小-最大。表中[2](#_bookmark29)我们对激活量子化提出了类似的比较。我们注意到，MSE结合最后一层的交叉熵，称为MSE+Xent，优于其他方法，特别是在较低的比特宽度下。该表还清楚地表明了使用交叉熵来代替MSE目标对最后一层的好处。

### 跨层均衡

量化误差的一个常见问题是，同一张量中的元素可能具有明显不同的大小。如上一节所讨论的，量化网格的范围设置试图在剪裁和舍入误差之间找到一个良好的折衷。不幸的是，在某些情况下，它们之间的幅度差异如此之大，以至于即使对于适度的量化（例如，INT8），我们也找不到合适的权衡。[Nagel等人（2019年](#_bookmark88))表明，这在深度可分离层中尤其普遍，因为每个输出特征只有少数权重负责，这可能导致权重的更高可变性。此外，他们还指出，批量规范化折叠会增加这种效果，并可能导致连接到各个输出通道的权重之间的严重不平衡（见图[5](#_bookmark32)）。虽然后者对于更细粒度的量化粒度(例如，每通道量化)来说不是一个问题，但是对于更广泛使用的每张量量化来说，这仍然是一个大问题。几篇论文（[Krishnamoorthi，2018年](#_bookmark85); [Nagel等人，2019](#_bookmark88); [盛等，2018 b](#_bookmark95)）注意到具有深度可分离卷积的高效模型，例如MobileNetV1（[霍华德](#_bookmark81)

每个输出通道的权重范围

100

75

50

25

0

25

50

1 2 3 4 5 6 7 8 9 1011121314151617181920212223242526272829303132

输出通道索引

图5：BN折叠后MobileNetV2中第一深度可分离层的每个（输出）通道权重范围。盒图显示每个通道的最小值和最大值、第二和第三四分位以及中值。

**W**(1) **W**(2)

**y**(2)

图6：单个通道的重新缩放说明。将第一层中的信道缩放因子s*i*导致第二层中的等效信道重新参数化1/s*i*。

[等，2017年](#_bookmark81)）和MobileNetV2（[桑德勒等人，2018](#_bookmark93)），显示PTQ显著下降，甚至导致随机性能。

本发明提出了一种无需使用每通道量化的克服这种不平衡的解决方案[Nagel等人（2019年](#_bookmark88)）。在并行工作中引入了类似的方法[梅勒等人（2019年](#_bookmark86)）。在这两篇论文中，作者观察到，对于许多常见的激活函数（例如，ReLU、PreLU），正标度等价性保持：

*f* (*sx*) = *sf* (*x*) (20)

对于任何非负实数s。这个等变系数适用于任何一阶齐次函数，并且可以通过缩放其参数化（例如ReLU6）来扩展到适用于任何分段线性函数。我们可以利用神经网络中连续层的正标度等价性。给定两层，h=f（W(1)x+b(1)）和y=f（W(2)h+b(2)），通过缩放等变，我们得到：

**y** = *f* (**W**(2)*f* (**W**(1)**x** + **b**(1)) + **b**(2))

=f(W(2)Sf(S−1W(1)x+S−1b(1))+b(2))

= *f* (**W**(2)*f*ˆ(**W**(1)**x** + **b**(1)) + **b**(2)) (21)

其中S=diag(S)是一个对角线矩阵，其值S*ii*表示神经元i的缩放因子S*i*。这允许我们用W(2)=W(2)S、W(1)=S−1W(1)和b(1)=S−1b(1)重新参数化我们的模型。在CNN的情况下，缩放将是每个频道，并相应地在空间维度上广播。我们在图中说明了这个重新缩放过程[6](#_bookmark34).

为了使模型对量化更稳健，我们可以找到一个缩放因子s*i*，使得重新缩放层中的量化噪声最小。跨层均衡（CLE）过程

|  |  |  |
| --- | --- | --- |
| 模型 | FP32 | INT8 |
| 原始模型 | 71.72 | 0.12 |
| + CLE | 71.70 | 69.91 |
| +吸收偏差 | 71.57 | **70.92** |
| 每通道量化 | 71.72 | 70.65 |

表3：MobileNetV2跨层均衡（CLE）的影响。ImageNet验证精度(%)，以全精度和8位量化进行评估。

([Nagel等人，2019](#_bookmark88)）通过在连续层上均衡动态范围来实现这一点。他们证明了通过设置S以使：

**s =**1 小(1)**r**(2)

(22)

其中r(j)是权张量j的通道i的动态范围。的算法[梅勒等人（2019年](#_bookmark86)）引入了一个类似的缩放因子，该因子也考虑了中间激活张量。然而，他们没有证明这种方法是最优的。

**吸收高偏压**[Nagel等人（2019年](#_bookmark88)）进一步注意，在某些情况下，特别是CLE之后，高偏压会导致激活的动态范围不同。因此，他们提出了一种程序，如果可能的话，将高偏差吸收到下一层。为了将c从第一层（随后是ReLU激活函数f）吸收到第二层，我们可以进行以下重新参数化：

**y** = **W**(2)**h** + **b**(2)

= **W**(2)(*f* (**W**(1)**x** + **b**(1)) + **c** − **c**) + **b**(2)

= **W**(2)(*f* (**W**(1)**x** + **b** (1)) + **c**) + **b**(2)

其中b(2)=W(2)c+b(2)，h=h c，和b(1)=b(1)c。在第二步中，我们利用这样一个事实，即对于具有ReLU函数f的层，存在一个非负向量c，使得r(Wx+Bc)=r(Wx+b)c。平凡解c=0对所有x都成立。然而，根据x的分布以及W和b的值，可以有一些值c*i*>0，对于这些值，这个等式对经验分布中的（几乎）所有x都成立。此值等于

c*i*=最大(0，最小(W(1)x+b(1)))。(24)

其中在小校准数据集中评估min**x**。为了消除对数据的依赖，作者提出了估计（[24](#_bookmark37)）通过批处理规范化层的移位和缩放参数，结果[3](#_bookmark0) 在c*i*=最大值(0，β I-3 γ I)。

**表中的实验**[3](#_bookmark35)我们证明了CLE和偏置吸收对将MobileNetV2量子化到8位的影响。由于跳过连接打破了层之间的等价性，我们只对每个剩余块内的层应用跨层等价。类似于[克里希纳穆尔蒂（2018](#_bookmark85)），我们观察到当将MobileNetV2量化到INT8时，模型性能接近随机。应用CLE使我们回到了FP32性能的2%以内，接近于每通道量化的性能。我们注意到，吸收高偏置会导致FP32性能的小下降，因为它是一种近似，但由于更精确的激活量化，它将量化性能提高了1%。CLE和偏置吸收之后的每张量量化共同产生比每通道量化更好的结果。

### 偏置校正

另一个常见的问题是量化误差往往有偏差。这意味着原始层或网络的预期输出和量化层或网络的预期输出发生了移位（E[Wx]=E Wx）。这种错误是

3 假设x为正态分布，则等式将占输入的约99.865%。

在深度可分离层中更为明显，每个输出通道只有几个元素（对于3 3内核通常为9个）。造成这种误差的主要原因往往是剪枝误差，因为少数强剪枝异常可能导致预期分布的变化。

几篇论文（[Nagel等人，2019](#_bookmark88); [Meller等人，2019](#_bookmark86); [芬克尔斯坦等人，2019](#_bookmark78)）注意到了这个问题，并介绍了针对预期的分布变化进行修正的方法。对于量化误差Δ W=W---W的量化层W，预期输出分布为

= E [(**W** + ∆**W**)**x**]

=E[Wx]+E[Δ wx]。(25)

因此，偏置误差由E[Δ wx]给出。因为Δ w是常数，所以我们有E[Δ wx]=Δ w e[x]。如果Δ w e[x]非零，则输出分布发生偏移。为了抵消这种变化，我们可以从输出中减去它：

E【ycorr】=E Wx−Δ W E【x】=E【y】。(26)

注意，这个修正项是一个与偏置形状相同的向量，因此可以被吸收到偏置中，而在推断时没有任何额外的开销。计算偏差修正项的方法有几种，其中最常见的两种是经验偏差修正和解析偏差修正。

经验偏差校正如果我们有一个校准数据集，偏差校正项可以通过比较量化模型和全精度模型的激活来简单地计算。实际上，这可以通过计算逐层完成

Δ W E[x]=E(Wxl-E[Wx]。(27)

**分析偏差校正**[Nagel等人（2019年](#_bookmark88)介绍了一种不需要数据就能解析计算偏置误差的方法。对于具有批量归一化和ReLU函数的常见网络，它们使用前一层的BN统计量来计算期望的输入分布E[x]。BN参数γ和β对应于BN层输出的平均值和标准差。假设输入值为正态分布，则可以使用剪枝正态分布来模拟ReLU对分布的影响。他们表明

E[x]=E[ReLU(x前)]

=γ N( −β }+β 1-φ( −β } (28)

其中x前是预激活输出，假定为每通道均值β和每通道标准差γ的正态分布，Φ()是标准正态CDF，符号(x)用于表示标准正态PDF。注意，所有向量操作都是元素（每个通道）操作。在计算输入分布E[x]后，将其乘以权量化误差Δ w即可简单地导出校正项。

**表中的实验**[4](#_bookmark40)我们演示了偏置校正对将MobileNetV2量化到8位的影响。应用解析偏差校正可以使量化模型的性能从随机提高到50%以上，表明量化引入的偏差误差对模型性能有显著的损害。当将偏置校正与CLE结合时，我们看到两种技术是互补的。它们一起在不使用任何数据的情况下实现了接近FP32的性能。

### 阿达罗恩

神经网络权值通常通过将每个FP32值投影到最近的量化网格点来量化，如方程中所示（[4](#_bookmark8)）用于均匀量化网格。我们将这种量化策略称为舍入到最近。舍入到最近策略的动机是这样一个事实，即对于固定量化网格，它产生浮点和量化权重之间的最低MSE。然而，[Nagel等人（2020](#_bookmark89)）表明，就

|  |  |  |
| --- | --- | --- |
| 模型 | FP32 | INT8 |
| 原始模型 | 71.72 | 0.12 |
| +偏置校正 | 71.72 | **52.02** |
| CLE+偏置吸收 | 71.57 | 70.92 |
| +偏置校正 | 71.57 | **71.19** |

表4：MobileNetV2的偏置校正效果。在全精度和8位量化下评估的ImageNet验证精度(%)。

65

60

55

50

45

40

35

0.5 1.0 1.5 2.0 2.5 3.0

1 wTH(w)w

2

图7：等式中成本之间的相关性（[30](#_bookmark44))与仅用于ResNet18第一层的4位量化的100随机舍入矢量w的ImageNet验证精度(%)相比。

训练后制度中量化权重时的任务损失。为了说明这一点，作者使用100个不同的随机舍入样本将ResNet18的第一层的权重量化到4位（[Gupta等人，2015年](#_bookmark79)）并评估了每个舍入选择的网络性能。其中最好的舍入选择比最接近舍入的表现高出10%以上。图[7](#_bookmark42) 通过绘制这些取整选择在y轴上的性能来说明这一点。在本节中，我们将描述AdaRound（[Nagel等人，2020](#_bookmark89)），一种为PTQ寻找良好的权重舍入选择的系统方法。AdaRound是一种理论上有充分依据且计算效率高的方法，在实践中显示出显著的性能改进。

由于量化的主要目标是最小化最终任务损失，因此我们首先从这个损失出发，制定优化问题

最小参数

∆**w**

E [L (**x***,* **y***,* **w** + ∆**w**) − L (**x***,* **y***,* **w**)] (29)

其中Δ W表示由于量化引起的扰动，并且可以为每个权重取两个可能的值，一个是通过将权重向上舍入，另一个是通过将权重向下舍入。我们想要高效地解决这个二元优化问题。作为第一步，我们使用二阶泰勒级数展开来近似成本函数。这就减少了在优化过程中对每个新舍入选择进行性能评估的需要。我们进一步假设模型是收敛的，这意味着梯度项在近似中的贡献可以忽略，并且Hessian是块对角线，这忽略了跨层相关性。这就导致了以下基于Hessian的二次无约束二进制优化问题

最小参数

∆**w**()

E (∆**w**()*T* **H**(**w**())∆**w**()l (30)

图中的明显相关性[7](#_bookmark42) 在验证精度和方程的目标之间（[30](#_bookmark44)）表示后者作为任务损失的良好代理（等式[29](#_bookmark43)），甚至用于4位权重量化。尽管业绩有所改善（见表[5](#_bookmark45)），等式（[30](#_bookmark44)）不能广泛应用于重量取整，主要有两个原因：

* 计算Hessian的内存和计算复杂性对于一般的用例是不实用的。

|  |  |  |
| --- | --- | --- |
| 四舍五入 | 第一层 | 所有层 |
| 最近 | 52.29 | 23.99 |
| H(w)任务损失(等式[30](#_bookmark44)) | 68.62 | N/A |
| 续。松弛MSE（方程[32](#_bookmark48)) | 69.58 | 66.56 |
| AdaRound（方程式[35](#_bookmark49)) | **69.58** | **68.60** |

表5：本节所作各种近似和假设的影响[3.4](#_bookmark39) 在ImageNet上，ResNet18的验证准确度(%)平均超过5次运行。不适用表示相应的实验在计算上是不可行的。

* 方程的曲波问题（[30](#_bookmark44)）是NP-硬的。

为了解决第一个问题，作者引入了额外的适当假设，允许简化方程的目标（[30](#_bookmark44)）到以下局部优化问题，该问题最小化了层的输出激活的MSE。

最小参数

()

*k,*:

(∆**W**()**x**(−1))2 (31)

方程（[31](#_bookmark47)）不需要计算海森系数，也不需要从后续层中计算任何其他向后或向前传播信息。请注意，用于将方程的QUBO问题联系起来的近似和分析（[30](#_bookmark44)）方程的局部优化问题（[31](#_bookmark47)）独立于舍入问题。因此，这种分析也有利于其他问题的算法设计，包括模型压缩和NAS([Moons等人，2020](#_bookmark87)).

优化（[31](#_bookmark47)）仍然是一个NP难的优化问题。为了找到一个计算复杂度合理的近似解，作者将优化问题放宽到以下连续优化问题

arg minIWx-WxI+λ f

(**V**) *,* (32)

**V** I I*F* reg

其中l·l2

表示Frobenius范数，W是定义为

=s·夹具( W+h(V)；n，p}。(33)

我们用n和p表示整数网格极限，n=qmin/s和p=q最大/s。V*i,j*是我们优化的连续变量，h可以是值在0到1之间的任何单调函数，即h（V*i,j*）[0,1]。在[Nagel等人（2020](#_bookmark89)），作者使用一个校正的乙状结肠作为h。（[32](#_bookmark48))还引入了一个正则项，该正则项鼓励连续优化变量h(V*i,j*)收敛到0或1，从而它们是离散优化的有效解([31](#_bookmark47)）。用于[Nagel等人（2020](#_bookmark89)）是

freg (V) = 1 − |2*h* (**V***i,j*) − 1|*β,* (34)

*i,j*

其中β在优化过程中退火，以最初允许h(V*i,j*)自由移动，然后迫使它们收敛到0或1。为了避免误差在神经网络中的跨层累积，并考虑到非线性，作者提出了以下最终优化问题

arg minIf(Wx)-f(Δ)I2+λ f(**V**) *,* (35)

**V** I *a a* I*F* reg

其中，T是对所有前面的层进行量化的层的输入，f*a*是激活函数。（[35](#_bookmark49)）可以使用随机梯度下降有效地优化。这种优化权重取整的方法被称为“环绕”。

综上所述，我们在量化运算过程中的权重舍入方式对网络的性能有重大影响。AdaRound提供了一种理论上合理、计算上快速的权重取整方法。它只需要少量的未标记数据样本，不需要超参数整定或端到端微调，可以应用于任何神经网络的全连接和卷积层。

图8：标准PTQ管道。蓝色方框表示所需步骤，绿松石方框表示推荐的选择。

### 标准PTQ管道

在本节中，我们根据相关文献和大量实验，提出了PTQ的最佳实践管道。我们在图中说明了推荐的管道[8](#_bookmark50).该管道可为许多计算机视觉以及自然语言处理模型和任务实现具有竞争力的PTQ结果。根据模型的不同，可能不需要某些步骤，或者其他选择可能导致相同或更好的性能。

跨层均衡首先，我们应用了跨层均衡（CLE），这是全精度模型的预处理步骤，使其对量化更加友好。CLE对于具有深度可分离层的模型和每张量量化特别重要，但它也经常显示出对其他层和量化选择的改进。

添加量化器接下来，我们选择我们的量化器并在我们的网络中添加量化操作，如第节所述[2.3](#_bookmark13)量化器的选择可能取决于特定的目标硬件；对于常见的人工智能加速器，我们建议使用对称量化器进行权重，使用非对称量化器进行激活。如果HW/SW堆栈支持，则对权值使用每通道量化是有利的。

权重范围设置为了设置所有权重张量的量化参数，我们建议使用基于分层MSE的准则。在特定的每通道量化情况下，使用min-max方法在某些情况下是有利的。

如果我们有一个小的校准数据集[4](#_bookmark0) 我们接下来将应用AdaRound，以优化权重的舍入。这一步骤对于在PTQ中实现低比特权重量化（例如4比特）至关重要。

偏差校正如果我们没有这样的校准数据集，而网络使用批量归一化，我们可以使用分析偏差校正来代替。

激活范围设置作为最后一步，我们确定网络中所有数据相关张量的量化范围（即激活）。我们对大多数层使用基于MSE的标准，这需要一个小的校准集来找到最小的MSE损耗。或者，我们可以使用基于BN的范围设置来拥有完全无数据的管道。

### 实验

我们现在评估上述PTQ管道在常见计算机视觉和自然语言理解应用中的性能。我们的结果总结于表[6](#_bookmark53).对于语义分割任务，我们评估了DeepLabV3（使用MobileNetV2主干）（[陈等，2017](#_bookmark74))在Pascal VOC上，对于对象检测，EfficientDet([谭等，2020](#_bookmark98)）在可可2017年。其余的计算机视觉模型在ImageNet分类基准上进行评估。对于自然语言理解，我们基于GLUE基准（[王等，2018](#_bookmark100)).

在所有情况下，我们观察到，与浮点（0.7%以内）相比，权重和激活的8位量化（W8A8）仅导致所有模型的精度边际损失。适用于W8A8

4 通常，500到1000个未标记图像就足以作为校准集。

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | FP32 | 每张量  W8A8 W4A8 | 每通道  W8A8 W4A8 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ResNet18 | 69.68 | 69.60 | 68.62 | 69.56 | 68.91 |
| ResNet50 | 76.07 | 75.87 | 75.15 | 75.88 | 75.43 |
| 移动电视2 | 71.72 | 70.99 | 69.21 | 71.16 | 69.79 |
| InceptionV3 | 77.40 | 77.68 | 76.48 | 77.71 | 76.82 |
| EfficientNet精简版 | 75.42 | 75.25 | 71.24 | 75.39 | 74.01 |
| DeeplabV3 | 72.94 | 72.44 | 70.80 | 72.27 | 71.67 |
| EfficientDet-D1 | 40.08 | 38.29 | 0.31 | 38.67 | 35.08 |
| BERT基† | 83.06 | 82.43 | 81.76 | 82.77 | 82.02 |

表6：我们的标准PTQ管道在各种型号和任务中的性能（平均超过5次运行）。DeeplabV3（MobileNetV2主干）在Pascal VOC（联合上的平均交叉点）、COCO 2017上的EfficientDet-D1（平均平均精度）、BERT——基于GLUE基准和ImageNet上的其他模型（精度）上进行评估。我们在各自的验证集中评估所有模型。在所有情况下，越高越好。†一些量化的激活保持在更高的精度(16位)。

量化我们也没有看到使用每通道量化的显著收益。然而，当权重量化到4位时，图像发生变化(W4A8)。对于ResNet18/50和InceptionV3，对于每张量和每通道量化，精度下降仍在浮点的1%以内。然而，对于更高效的网络，如MobileNetV2和EfficientNet lite，对于每张量量化，下降幅度分别增加到2.5%和4.2%。这很可能是由于深度可分离卷积的量化。这里，每通道量化可以显示出显著的好处，例如，在EfficientNet lite中，与每张量量化相比，每通道量化将精度提高2.8%，使其在1.4%的全精度精度内。我们看到了EfficientDet-D1和DeeplabV3的类似效果，它们都在主干中使用深度可分离卷积。

对于BERT-base，我们观察到一些激活张量在其动态范围上存在极端的差异。为了使PTQ仍然工作，我们使用第#节中概述的调试过程识别了这些层[3.7](#_bookmark54) 并将它们保存在16位。否则，BERT-base遵循与大多数其他型号相似的趋势，我们的PTQ管道允许4位权重量化在1.5%的胶分数下降。

### 调试

我们证明了标准PTQ管道可以在广泛的模型和网络中获得具有竞争力的结果。但是，如果在遵循我们的管道步骤后，模型的性能仍然不令人满意，我们建议使用一组诊断步骤来识别瓶颈并提高性能。虽然这不是严格意义上的算法，但这些调试步骤可以提供关于量化模型性能不佳的见解，并有助于解决潜在问题。这些步骤如图所示[9](#_bookmark55) 并在下文中更详细地描述：

FP32的合理性检查一个重要的初始调试步骤是确保浮点和量化模型在前向传递中的行为相似，尤其是在使用自定义量化管道时。将量化模型位宽设置为32位，用于权重和激活，或在可能的情况下绕过量化操作，并检查精度是否与FP32模型匹配。

权重或激活量化下一个调试步骤是确定激活或权重量化如何独立地影响性能。如果所有权重都量化到较高的位宽，而激活保持在较低的位宽，或者如果所有激活都使用高位宽而激活低位宽，性能是否恢复？这一步可以显示激活和权重量化对整体性能下降的相对贡献，并为我们指出合适的解决方案。

如果上一步显示权重量化确实会导致精度下降，那么有几种解决方案可以尝试：

* 如果尚未实现CLE，则应用CLE，特别是对于具有深度可分离卷积的模型。

图9：PTQ调试流程图。误差是浮点和量化模型精度之间的差值。

* 尝试每通道量化。这将解决每个信道权重分配不均的问题。
* 如果有校准数据，则应用偏置校正或在周围应用。

固定激活量化以减少激活量化带来的量化误差，我们也可以尝试使用不同的范围设置方法或调整CLE以考虑激活量化范围，因为香草CLE会导致激活分布不均匀。

逐层分析如果全局解决方案的精度没有恢复到可接受的水平，我们将单独考虑每个量化器。我们将每个量化器按顺序设置为目标位宽，同时将网络的其余部分保持在32位（参见图中的inner for循环[9](#_bookmark55)).

可视化图层如果单个张量的量化导致精度显著下降，我们建议可视化不同粒度的张量分布，例如，如图所示的每个通道[5](#_bookmark32)和维度，例如，用于BERT中激活的每个令牌或每个嵌入。

固定单个量化器可视化步骤可以揭示张量对量化的敏感性的来源。一些常见的解决方案包括为该量化器设置自定义范围，或为有问题的量化器允许更高的位宽，例如，BERT-base from table[6](#_bookmark53)如果问题得到解决，并且精度恢复，我们将继续下一个量化器。如果没有，我们可能不得不求助于其他方法，例如量化感知训练（QAT），这在本节中讨论[4](#_bookmark57).

完成上述步骤后，最后一步是将完整模型量化到所需的位宽。如果精度是可接受的，我们就有了最终的量化模型。否则，我们可以考虑更高的位宽和更小的粒度，或者恢复到更强大的量化方法，例如量化感知训练。

# 量化感知训练

上一节中描述的训练后量化技术是我们量化工具包中的第一个常用工具。它们非常有效且快速实现，因为它们不需要对带有标签数据的网络进行重新训练。然而，它们有局限性，特别是当目标是低比特量化激活时，例如4比特及以下。训练后技术可能不足以减轻低比特量化引起的大量化误差。在这些情况下，我们采用量化感知训练（QAT）。QAT对量化噪声源进行建模（参见第[2.3](#_bookmark13)）在训练期间。这使得模型能够找到比训练后量化更多的最优解。然而，较高的精确度伴随着神经网络训练的通常成本，即较长的训练时间、对标记数据的需要和超参数搜索。

在本节中，我们将探讨反向传播如何在具有模拟量化的网络中工作，并为使用QAT有效地训练模型提供标准管道。我们还将讨论批量归一化、折叠和每通道量化在QAT中的含义，并为广泛的任务和模型提供结果。

### 反向路径的模拟量化

在节中[2.3](#_bookmark13)，我们看到了如何在深度学习框架中使用浮点来模拟量化。但是，如果我们看一下图的计算图[4](#_bookmark14)为了训练这样的网络，我们需要通过模拟的量化器块进行反向传播。这就产生了一个问题，因为方程中的舍入到最近操作的梯度（[4](#_bookmark8)）是零或到处都是未定义的，这使得基于梯度的训练变得不可能。一种解决方法是使用直通估计器(STE，[本吉奥等人。2013年](#_bookmark72)），将舍入运算符的梯度近似为1：

*∂lyl= 1* (36)

*∂y*

利用这个近似，我们现在可以从方程（[7](#_bookmark9)）。为了清楚起见，我们假设对称量子化，即z=0，但同样的结果也适用于非对称量子化，因为零点是一个常数。我们用n和p来定义整数网格极限，使得n=qmin/s和p=q最大/s。方程的梯度（[7](#_bookmark9))w.r.t其输入，

x*i*，由以下方式给出：

*∂***x**.*i* = *∂q*(**x***i*)

*∂***x***i*

*∂***x***i*

=s·*∂*夹具( xi 1；n，p)+0

*s ·∂lxi/sl∂(xi/s)*

*如果q*min*≤xi≤q*最大*，*

= 



*∂*(**x***i/s*)

*∂n s* · *∂***x**

*∂***x***i*

if x*i*

*< q*min*,*

*1如果q*min*≤xi≤q*最大*，*

0否则。

(37)

使用这个梯度定义，我们现在可以通过量化块反向传播。图[10](#_bookmark60) 显示了用于量化感知训练的前向和后向传递的简单计算图。前向通道与图中相同[4](#_bookmark14)，但在后向传递中，由于STE假设，我们有效地跳过了量化器块。在早期的QAT工作中，量化范围

 向后向前

输入

图10：采用STE假设的量化感知训练的正向和反向计算图。

对于在每次迭代中更新的权重和激活，最常用的是最小-最大范围（[Krishnamoorthi，2018年](#_bookmark85)）。在后来的工作中（[Esser等人，2020](#_bookmark77); [Jain等人，2019](#_bookmark84); [Bhalgat等人，2020](#_bookmark73)用STE计算梯度w.r.t。量化参数z和s。首先利用链式规则和STE计算梯度w.r.t。比例因子：

*∂***x**.*i* = *∂*

(s·夹具( xi 1；n，p)l

*-xi/s+lxi/sl如果q*min*≤xi≤q*最大*，*

= *n 如果xi<q*min*，*

*p 如果xi>q*最大*。*

(38)

最初，我们将零点限制为整数。为了使零点可学习，我们将其转换为实数并应用舍入算子。修改后的量化函数定义为：

**x** = *q*(**x**; *s, z*) = *s* 夹具 **x** + *z* ; *n, p z* (39)

*s*

梯度w.r.t。通过将STE再次应用于舍入运算符，计算到z：

*∂***x**.*i* = (0 *q*min *≤ xi ≤ q*最大*,*

(40)

|  |  |  |
| --- | --- | --- |
| 型号（FP32精度） | ResNet18 (69.68) | 移动电视2（71.72） |

|  |  |  |
| --- | --- | --- |
| 位宽 | W4A8 W4A4 | W4A8 W4A4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 静态折叠BN | **69.76** | **68.32** | **70.17** | **66.43** |
| 双前锋([Krishnamoorthi，2018年](#_bookmark85)) | 69.42 | 68.20 | 66.87 | 63.54 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 静态折叠（每个通道） | 69.58 | 68.15 | **70.52** | 66.32 |
| 保留原始BN（每个频道） | **70.01** | **68.83** | 70.48 | **66.89** |

表7：将BN纳入QAT的各种方法的消融研究。学习速率针对每个配置单独优化。3次运行的平均ImageNet验证准确率(%)。

### 批量归一化折叠和QAT

在节中[2.3.1](#_bookmark16)我们引入了批量归一化折叠，将缩放和加法吸收到线性层中，以允许更有效的推理。在量化感知训练中，我们想要紧密模拟推理行为，这也是为什么我们在训练中要考虑BN折叠的原因。注意，在一些QAT文献中，BN折叠效应被忽略了。虽然在我们使用每通道量化时（本节下文将详细介绍）这是可以的，但是保持BN展开以进行每张量量化会导致以下两种情况之一：

1. BN层在推理期间应用每个信道的重新缩放。在这种情况下，我们不妨首先使用每通道量化。
2. 我们在部署过程中将BN折叠到权张量中，并且在训练网络以适应不同的量化噪声时，会导致潜在的显著精度下降。

在QAT中建模BN折叠的一个简单而有效的方法是静态折叠BN尺度并将其偏移到线性层的权重和偏差中，正如我们在方程中看到的那样([11](#_bookmark21)）和（[12](#_bookmark22)）。这对应于权重的重新参数化，并且有效地从网络中完全去除批量规范化操作。从一个收敛的预训练模型开始，静态折叠是非常有效的，从表的结果可以看出[7](#_bookmark61).

另一种方法是[雅各布等人（2018年](#_bookmark83))两者都在QAT期间更新运行统计信息，并使用校正应用BN折叠。这种方法比较麻烦，计算成本也比较高，因为它涉及到一个双前向传递：一个用于批量统计，一个用于量化线性操作。然而，根据我们的实验（见表[7](#_bookmark61)），静态折叠尽管简单，但性能还是很好。

**分段每通道量化**[2.4.2](#_bookmark24)，我们提到权重的每通道量化在硬件支持下可以提高准确率。静态折叠重参数化也适用于每通道量化。然而，每通道量化提供了额外的灵活性，因为它允许我们将批量归一化缩放操作吸收到每通道缩放因子中。让我们通过从第1节重温BN折叠方程来了解这是如何可能的[2.3.1](#_bookmark16)，但这次引入了权重的每通道量化，使得W*k,:*=q(W*k,:*；s**w,k**)=

s**w,k**W国际。通过对类似于等式的线性层的输出应用批规范化([10](#_bookmark19)),

我们得到：

y*k*=批次标准(--k,：x)

***γ* W**

= ***σ***2 + *E* **x** +

***γ Kµkβ K−σ***2***+E***

***γ****ks***w***,k* 国际

= 2 **W***k,*: **x** + **b***k*

*k*

*k,*:

我们可以看到，现在可以将批规范化缩放参数吸收到每个通道缩放因子中。对于QAT来说，这意味着我们可以在训练时保持BN层的完整，并在之后将BN缩放因子合并到每个信道量化参数中。实际上，

|  |  |  |
| --- | --- | --- |
| 型号（FP32精度） | ResNet18 (69.68)  PTQ QAT | 移动电视2（71.72）  PTQ QAT |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| W4A8 w/min-最大重量初始化 | 0.12 | 69.61 | 0.56 | 69.96 |
| W4A8带MSE重量初始化 | 18.58 | **69.76** | 12.99 | **70.13** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| W4A4 w/最小-最大动作初始化 | 7.51 | 68.23 | 0.22 | **66.55** |
| W4A4带MSE动作初始化 | 9.62 | **68.41** | 0.71 | 66.29 |

表8：各种初始化量化网格方法的烧蚀研究。学习速率针对每个配置单独优化。ImageNet验证准确率（%）平均超过3次运行。

从表的最后两行可以看出，这种建模方法对于每通道量化来说是完全正确的，或者比静态折叠更好[7](#_bookmark61).

### QAT的初始化

在本节中，我们将探讨初始化对QAT的影响。文献中通常的做法是从预先训练的FP32模型开始([Esser等人，2020](#_bookmark77); [Krishnamoorthi，2018年](#_bookmark85); [雅各布等人，2018](#_bookmark83); [Jain等人，2019](#_bookmark84)）。虽然从FP32模型开始是有益的，但量化初始化对最终QAT结果的影响研究较少。在这里，我们探讨了在进行QAT之前使用几种PTQ技术的效果。

范围估计的影响，以评估初始范围设置的影响（参见第[3.1](#_bookmark28)对于重量和活化，我们进行了两组实验，总结如下[8](#_bookmark64)在第一个实验中，我们将权值量化到4位，并将激活保持在8位。我们比较了最小-最大初始化和基于MSE的权值量化范围的初始化。虽然MSE初始化模型具有显著更高的启动精度，但在训练20个时代后，差距缩小了。

为了探索激活量化的相同效果，我们进行了一个类似的实验，现在我们将激活量化到4位，并比较最小-最大初始化与基于MSE的初始化。权重范围初始化的观察结果也适用于此。图中[11](#_bookmark66) 我们展示了这个实验的完整训练曲线。在最初的几个时代，使用MSE初始化有一个显著的优势，在训练的后期几乎消失了。总之，更好的初始化可以导致更好的QAT结果，但增益通常很小，并且随着训练持续时间的延长而消失。

69

68

67

66

65

64

63

62

0 2 4 6 8 10 12 14 16 18 20

纪元

图11：初始激活范围设置对ResNet18 QAT训练行为的影响。每次训练周期后超过3次运行的平均ImageNet验证准确率（%）（标准偏差着色）。

|  |  |  |
| --- | --- | --- |
| 型号（FP32精度） | ResNet18 (69.68)  PTQ QAT | 移动电视2（71.72）  PTQ QAT |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| W4A8基线 | 18.58 | 69.74 | 0.10 | 0.10 |
| W4A8带CLE | 16.29 | **69.76** | 12.99 | **70.13** |
| W4A8带CLE+BC | 38.58 | 69.72 | 46.90 | 70.07 |

表9：不同PTQ初始化的消融研究。学习速率针对每个配置单独优化。ImageNet验证准确率（%）平均超过3次运行。

**CLE在表中的影响**[9](#_bookmark67) 我们比较了其他PTQ改进的效果，如CLE和偏置校正。对于ResNet18，我们没有看到最终QAT性能的显著差异，而对于MobileNetV2，我们观察到没有CLE就无法训练它。这可能是由于每张量量化导致的灾难性性能下降，我们在本节中讨论了这一点[3.2](#_bookmark31)总之，对于那些存在严重问题的模型，我们可能需要高级PTQ技术，如CLE来初始化QAT。在大多数其他情况下，改进的PTQ初始化只会导致最终QAT性能的轻微改善。

### 标准QAT管道

在本节中，我们根据相关文献和大量实验，提出了QAT的最佳实践管道。我们在图中说明了推荐的管道[12](#_bookmark68).该管道在多种计算机视觉和自然语言处理模型和任务上产生了良好的QAT结果，可以被视为实现低比特量化性能的首选工具。正如前面几节所讨论的，我们总是从一个预先训练的模型开始，并遵循一些PTQ步骤，以便具有更快的收敛速度和更高的精度。

与PTQ类似的跨层均衡，首先将CLE应用于全精度模型。正如我们在桌子上看到的[9](#_bookmark67)，对于遭受权重分布不平衡的模型，如MobileNet架构，此步骤是必要的。对于其他网络或在每信道量化的情况下，该步骤可以是可选的。

添加量化器接下来，我们选择我们的量化器并在我们的网络中添加量化操作，如第1节所述[2.3](#_bookmark13)量化器的选择可能取决于特定的目标硬件，对于常见的AI加速器，我们建议使用对称量化器进行权重，使用非对称量化器进行激活。如果HW/SW堆栈支持，则对权值使用每通道量化是有利的。在此阶段，我们还将确保批量规范化的模拟是正确的，如第#节所述[4.2](#_bookmark63).

在训练前，我们必须初始化所有的量化参数。更好的初始化将有助于更快的训练，并可能提高最终的准确性，尽管通常提高幅度很小（见表[8](#_bookmark64)）。一般来说，我们建议使用基于分层MSE的标准来设置所有量化参数。在特定的每通道量化情况下，使用最小-最大设置有时是有利的。

可学习量化参数我们建议使量化器参数可学习，如第#节所述[4.1](#_bookmark58)直接学习量化参数，而不是每次都更新，可带来更高的性能，尤其是在处理低比特量化时。但是，在为任务设置优化器时，使用可学习量化器需要特别小心。当使用SGD型优化器时，与网络的其余参数相比，量化参数的学习速率需要降低。如果我们使用具有自适应学习速率的优化器，例如Adam或RMSProp，则可以避免学习速率的调整。

### 实验

使用我们的QAT管道，我们量化和评估了与我们在部分中用于PTQ的相同模型[3.6](#_bookmark52)我们的结果见表[10](#_bookmark70) 用于不同的位宽和量化粒度。DeepLabV3在帕斯卡VOC上训练了80个时代；在COCO 2017上有效运行20个时代；所有其他视觉模型都在ImageNet上训练了20个时代。BERT-base在每个

图12：标准量化感知训练管道。蓝色方框表示步骤，绿松石方框表示推荐的选择。

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | FP32 | 每张量  W8A8 W4A8 W4A4 | 每通道  W8A8 W4A8 W4A4 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ResNet18 | 69.68 | 70.38 | 69.76 | 68.32 | 70.43 | 70.01 | 68.83 |
| ResNet50 | 76.07 | 76.21 | 75.89 | 75.10 | 76.58 | 76.52 | 75.53 |
| InceptionV3 | 77.40 | 78.33 | 77.84 | 77.49 | 78.45 | 78.12 | 77.74 |
| 移动电视2 | 71.72 | 71.76 | 70.17 | 66.43 | 71.82 | 70.48 | 66.89 |
| EfficientNet精简版 | 75.42 | 75.17 | 71.55 | 70.22 | 74.75 | 73.92 | 71.55 |
| DeeplabV3 | 72.94 | 73.99 | 70.90 | 66.78 | 72.87 | 73.01 | 68.90 |
| EfficientDet-D1 | 40.08 | 38.94 | 35.34 | 24.70 | 38.97 | 36.75 | 28.68 |
| BERT基 | 83.06 | 83.26 | 82.64 | 78.83 | 82.44 | 82.39 | 77.63 |

表10：我们的标准QAT管道在各种型号和任务中的性能（平均超过3次运行）。DeeplabV3（MobileNetV2主干）在Pascal VOC（联合上的平均交叉点）上进行评估，在COCO 2017上进行EfficientDet-D1（平均平均精度）上进行评估，在GLUE基准上进行BERT-base，在ImageNet上进行所有其他模型（精度）。我们在各自的验证集中评估所有模型。在所有情况下，越高越好。

根据任务和量化粒度，对应的胶水任务为3到12个时期。我们对所有模型都使用Adam优化器。我们以每个量化配置的最佳学习率呈现结果，并且不进行进一步的超参数调整。

我们观察到，对于没有深度可分离卷积的网络（表的前3行[10](#_bookmark70)W8A8和W4A8量化在某些情况下与浮点模型相当甚至优于浮点模型。这可能是由于量化噪声训练的规律性效应，或者是由于QAT期间的额外微调。对于更激进的W4A4，我们注意到一个小的下降，但仍然在1%的浮点精度。

具有深度可分离层的量化网络（MobileNetV2、EfficientNet lite、DeeplabV3、EfficientDet-D1）更具挑战性；我们还从本节的PTQ结果中观察到了一个趋势[3.6](#_bookmark52) 并在文献中讨论（[Chin等人，2020](#_bookmark75); [盛等，2018年a](#_bookmark94)）。8位量化几乎不会导致精度下降，而将权重量化到4位会导致更大的下降，例如对于采用每张量量化的EfficientNet lite，下降约4%。每通道量化可以显著提高性能，使DeepLabV3达到浮点精度，并将MobileNetV2和EfficientNet lite之间的差距缩小到1.5%以下。对于这种网络来说，将权值和激活量化到4位仍然是一个挑战，即使采用每通道量化，它也会导致高达5%的下降。效率DET-D1仍然比该组中的其他网络更难量化。

对于BERT-base，我们观察到具有范围学习的QAT可以有效地处理高动态范围，允许将所有激活保持在8位（与PTQ不同）。W4A8保持在原胶分数的1%以内，说明低比特权量化对于变压器型号来说不是问题。我们只注意到当将此与低比特激活量化（W4A4）结合时，性能显著下降。

# 摘要和结论

深度学习已经成为许多机器学习应用中不可或缺的一部分，现在可以在无数的电子设备和服务中找到，从智能手机和家用电器到无人机、机器人和自动驾驶汽车。随着深度学习在我们日常生活中的普及和普及程度的提高，对快速高效的神经网络推理的需求也在增加。神经网络量化是降低神经网络在推理过程中能量和时延要求的最有效方法之一。

量化使我们能够从浮点表示转换为定点格式，并且结合利用高效定点操作的专用硬件，有可能实现显著的功率增益和加速推理。然而，为了利用这些节省，我们需要稳健的量化方法，能够保持高精度，同时减少权值和激活的比特宽度。为此，我们考虑了两类主要的量化算法：训练后量化（PTQ）和量化感知训练（QAT）。

训练后量化技术采用预先训练的FP32网络，并将其转换为不需要原始训练管道的固定点网络。这使得它们成为一种轻量级的、按下按钮的量化方法，工程工作量和计算成本都很低。我们描述了PTQ的一系列最新进展，并介绍了一个PTQ管道，该管道可为广泛的模型和机器学习任务带来接近浮点的精度结果。特别是，使用所提出的管道，我们可以实现8位量化的权值和激活在所有网络的浮点精度仅1%。我们进一步证明，许多网络甚至可以量化到4位权值，而性能只会有很小的额外下降。此外，我们引入了一个调试工作流，以有效地识别和修复在量化新网络时可能出现的问题。

量化感知训练通过模拟量化操作对训练过程中的量化噪声进行建模。与PTQ相比，该训练程序允许找到更好的解决方案，同时实现更有效和更积极的激活量化。与PTQ类似，我们引入了一个标准的训练管道，利用了该领域的最新算法。我们还特别关注了QAT期间的批量规范化折叠，并表明简单静态折叠优于其他计算成本更高的方法。我们证明了我们的QAT管道可以实现4位权重量化，对于某些模型甚至可以实现4位激活，与浮点相比，精度只有很小的下降。

PTQ和QAT之间的选择取决于应用程序的精度和功率要求。这两种方法都是任何模型效率工具包的重要组成部分，我们希望我们提出的管道将帮助工程师以更少的时间和精力部署高性能的量化模型。

# 参考文献

Banner,R.,Nahshan,Y.和Soudry,D.后训练4位量子化卷积网络的快速部署。神经信息处理系统（NeuRIPS），2019年。[9](#_bookmark27)

Bengio，Y.，Léonard，N.和Courville，a.通过随机神经元估计或传播梯度进行条件计算。arXiv预印本arXiv：1308.3432，2013。[19](#_bookmark56)

Bhalgat，Y.，Lee，J.，Nagel，M.，Blankevoort，T.和Kwak，n.LSQ+：通过可学习的偏移和更好的初始化来改进低比特量化。2020年6月IEEE/CVF计算机视觉和模式识别会议（CVPR）研讨会论文集。[20](#_bookmark59)

Chen，L.-C.，Papandreou，G.，Schroff，F.和Adam，H.语义图像分割的Rethinking atrous卷积，2017。[16](#_bookmark51)

陈天伟、庄、毕义杰、钱德拉、V.和马库列斯库。一个加权位宽来控制它们。在巴托利和富西洛（编辑。），计算机视觉-ECCV 2020研讨会，第85-103页，Cham，2020。斯普林格国际出版公司。ISBN 978-3-030-68238-5。[24](#_bookmark69)

董，Z.，姚，Z.，Gholami，A.，Mahoney，M.W.和Keutzer，K.Hawq：混合精度神经网络的hessian感知量化。国际计算机视觉会议（ICCV），2019年。[8](#_bookmark23)

Esser，S.K.，McKinstry，J.L.，Bablani，D.，Appuswamy，R.和Modha，D.S.学习了步长量化。国际学习表征会议（ICLR），2020年。[20](#_bookmark59), [22](#_bookmark65)

Finkelstein，a.，Almog，U.和Grobman，M.用偏差对抗量化偏差。arXiv预印本arXiv：1906.03193，2019。[13](#_bookmark38)

古普塔、阿格拉瓦尔、戈帕拉克里希南和纳拉亚南。数值精度有限的深度学习。国际机器学习会议，ICML，2015年。[14](#_bookmark41)

Horowitz，M.1.1计算的能量问题（以及我们可以做些什么）。2014年IEEE国际固态电路会议技术论文摘要（ISSCC），第10-14页，2014年。doi：10.1109/ISSCC.2014.6757323。[2](#_bookmark1), [3](#_bookmark3)

Howard，A.G.，Zhu，M.，Chen，B.，Kalenichenko，D.，Wang，W.，Weyand，T.，Andreetto，M.和Adam，H.MobileNets：用于移动视觉应用的高效卷积神经网络。arXiv预印本arXiv：1704.04861，2017。[10](#_bookmark30)

批规范化：通过减少内部协变量移位来加速深度网络训练。在巴赫和布莱（编辑。），第32届国际机器学习会议论文集，《机器学习研究论文集》第37卷，第448-456页，法国里尔，2015年7月7日至09日。PMLR.URL<http://proceedings.mlr.press/v37/ioffe15.html>.

[6](#_bookmark15)

雅各布、克里吉斯、陈、朱、唐、霍华德、亚当、卡列尼琴科，

D.神经网络的量化和训练，用于有效的整数-仅算术推理。

*计算机视觉与模式识别会议（CVPR），2018年。*[3](#_bookmark3), [6](#_bookmark15), [21](#_bookmark62), [22](#_bookmark65)

Jain,S.R.,Gural,A.,Wu,M.和Dick,C.训练了统一量化，用于在定点硬件上精确高效地进行神经网络推理。CoRR，abs/1903.08066，2019。URL[Tools](http://arxiv.org/abs/1903.08066)

[//arxiv.org/abs/1903.08066](http://arxiv.org/abs/1903.08066). [20](#_bookmark59), [22](#_bookmark65)

Krishnamoorthi，R.量化深度卷积网络以实现有效推理：白皮书。arXiv预印本arXiv：1806.08342，2018。[6](#_bookmark15), [10](#_bookmark30), [12](#_bookmark36), [20](#_bookmark59), [21](#_bookmark62), [22](#_bookmark65)

梅勒，E.，芬克尔斯坦，a.，阿尔莫格，U.和格罗布曼，M.相同，相同但不同：通过权因子法恢复神经网络量化误差。载于第36届机器学习国际会议论文集，ICML 2019，2019年6月9日至15日，美国加利福尼亚州长滩，第4486-4495页，2019年。URL<http://proceedings.mlr.press/v97/meller19a.html>. [11](#_bookmark33), [12](#_bookmark36), [13](#_bookmark38)

Moons，B.，Noorzad，P.，Skliar，a.，Mariani，G.，Mehta，D.，Lott，C.，和Blankevoort，T.蒸馏最佳神经网络：在不同空间中的快速搜索。arXiv预印本arXiv：2012.08859，2020。[15](#_bookmark46)

Nagel，M.，van Baalen，M.，Blankevoort，T.，和Welling，M.通过权重均衡和偏差校正的无数据量化。国际计算机视觉会议（ICCV），2019年。[10](#_bookmark30), [11](#_bookmark33), [12](#_bookmark36), [13](#_bookmark38)

内格尔，M.，阿姆贾德，R.A.，范巴伦，M.，路易索斯，C.和布兰克沃特，T.。向上还是向下？用于训练后量化的自适应舍入。在第三，H.D.和Singh，A.（编辑。），第37届国际机器学习会议论文集，机器学习研究论文集第119卷，第7197-7206页。PMLR，2020年7月13日至18日。URL[http://proceedings.mlr.press/](http://proceedings.mlr.press/v119/nagel20a.html)

[v119/nagel20a.html](http://proceedings.mlr.press/v119/nagel20a.html). [13](#_bookmark38), [14](#_bookmark41), [15](#_bookmark46)

Nascimento，M.G.D.，Fawcett，R.和Prisacariu，V.A.Dsconv：有效卷积算子。IEEE/CVF国际计算机视觉会议（ICCV）会议录，2019年10月。[5](#_bookmark10)

拉马钱德兰，P.，佐普，B.和勒，Q.V.正在搜索激活函数。CoRR，abs/1710.05941，2017。URL<http://arxiv.org/abs/1710.05941>. [7](#_bookmark17)

鲁哈尼、罗、赵、刘、米、福沃斯、奥夫恰洛夫、维诺格拉茨基、马森吉尔、杨、比特纳、福林、朱、何、纳、T、帕特尔、P、车、科帕卡、L.C.、宋、X、索姆、S、达斯、K、蒂瓦里、S、莱因哈特、S、兰卡、S、钟、E、伯格，用微软浮点在云尺度上突破了窄精度推理的极限。在神经信息处理系统中(NeurIPS 2020)。ACM，2020年11月。[5](#_bookmark10)

桑德勒、霍华德、朱、日莫吉诺夫和陈。Mobilenetv2：倒置残差和线性瓶颈。IEEE计算机视觉和模式识别会议论文集，第4510-4520页，2018年。[11](#_bookmark33)

盛，冯，卓，张，申，林，张一种用于移动网络的量化友好的可分离卷积。2018年第一届面向嵌入式应用的节能机器学习和认知计算研讨会(EMC2)，2018年3月a。doi：10.1109/emc2.2018。00011.URL<http://dx.doi.org/10.1109/EMC2.2018.00011>. [24](#_bookmark69)

盛，冯，卓，张，申，林，张一种用于移动网络的量化友好的可分离卷积。第一届嵌入式应用节能机器学习和认知计算研讨会(EMC2)，2018b。URL[https://ieeexplore.ieee.org/ abstract/document/8524017](https://ieeexplore.ieee.org/abstract/document/8524017). [10](#_bookmark30)

斯托克，P.，朱林，a.，格里邦瓦尔，R.，格雷厄姆，B.和杰古，H.然后，这一步就走了：重新审视神经网络的量化。CoRR，abs/1907.05686，2019。URL[http://arxiv.org/abs/ 1907.05686](http://arxiv.org/abs/1907.05686). [5](#_bookmark10)

谭，庞，乐，Efficientdet：可扩展和高效的对象检测，2020。[16](#_bookmark51) Uhlich，S.，Mauch，L.，Cardinaux，F.，Yoshiyama，K.，Garcia，J.A.，Tiedemann，S.，Kemp，T.，和

中村混合精度DNN：您只需要一个良好的参数化。2020年国际学习表征会议。URL[https://openreview.net/forum?id= Hyx0slrFvH](https://openreview.net/forum?id=Hyx0slrFvH). [8](#_bookmark23)

范·巴伦，M.，路易斯，C.，内格尔，M.，阿姆贾德，R.A.，王，Y.，布兰克沃特，T.和威灵，M.贝叶斯比特：统一量化和剪枝。在拉罗谢尔，H.，兰扎托，M.，哈德塞尔，R.，巴尔坎，M.F.和林，H.（编辑。），神经信息处理系统的进展，第33卷，第5741-5752页。柯伦联营公司，2020年。URL[https://proceedings.neurips.cc/ paper/2020/file/3f13cf4ddf6fc50c0d39a1d5aeb57dd8-Paper.pdf](https://proceedings.neurips.cc/paper/2020/file/3f13cf4ddf6fc50c0d39a1d5aeb57dd8-Paper.pdf). [8](#_bookmark23)

王，A.，辛格，A.，迈克尔，J.，希尔，F.，利维，O.，鲍曼，S.胶水：一个自然语言理解的多任务基准和分析平台。2018年EMNLP研讨会会议录BlackboxNLP：分析和解释NLP的神经网络，第353-355页，比利时布鲁塞尔，2018年11月。计算语言学协会。doi：10.18653/v1/W18-5446。URL<https://www.aclweb.org/anthology/W18-5446>. [16](#_bookmark51)