三值量化：综述

Dan Liu，**Xue Liu**麦吉尔大学
daniel.liu@mail.mcgill.ca,
xueliu@cs.mcgill.ca

arXiv:2303.01505v1 [cs.LG] 2 Mar 2023

摘要

推理时间、模型大小和准确率对于部署深度神经网络模型至关重要。为实现更快的推理速度和更高的准确率，研究人员开展了大量工作来压缩神经网络模型。剪枝和量化是实现这一目标的主流方法。在模型量化过程中，将层权重中的各个浮点值转换为低精度值，可以大幅降低计算开销并提高推理速度。已有许多量化方法得到研究，例如矢量量化、低比特量化以及二值／三值量化。本文聚焦于三值量化，回顾其发展历程，并从投影函数和优化方法的角度探讨现有三值量化方法之间的关系。

1 引言

深度神经网络（DNN）模型包含数百万个参数。在延迟、内存和功耗限制下部署 DNN 模型时，模型大小和推理效率是主要挑战。DNN 模型量化极具吸引力，因为它可以通过降低位宽，加速训练和推理过程并减少资源消耗。尽管有这些优势，量化也带来了挑战，尤其是在训练效率和性能方面：i）与传统方法相比，优化需要付出更多努力，训练时间也更长；ii）在大型数据集上训练的模型通常需要先进行预训练，然后再量化。

量化最显著的缺点是，量化模型的准确率低于其全精度模型对应版本：位宽越低，这种差异就越明显。二值量化和三值量化是两个极端情况，可以将乘法运算完全替换为结合硬件实现的位移运算。与其他更高位宽的量化工作（2 至 16 位）相比，这能够带来可观的能效提升。然而，采用二值量化时，很难保持较高的准确率。三值模型可以以较小的额外位宽成本，实现显著更好的性能，其性能可与全精度模型相媲美。此外，与二值量化不同，将对称的更高位宽量化转换为三值量化相对容易。换言之，三值量化是现有更高位宽（2–16 位）对称量化工作中的一种特殊情况。因此，对三值量化相关工作的总结将大有裨益。本综述聚焦于三值权重量化。有关二值量化及其他量化工作的详细介绍，请参见 Amir 等人的工作等 [2]。

量化可以通过多种方式分类：均匀、非均匀、对称、非对称、训练后量化和量化感知方法。一般而言，量化方法包含两个步骤：阈值化和投影。因此，本综述从投影函数和优化方法两个方面总结三值量化。投影函数根据量化策略进行分类。优化方法则根据使用梯度优化模型参数的方式进行分类。我们使用近端算子 [37]揭示现有研究中各种优化方法之间的内在联系。

本综述的主要贡献如下：

- 全面且深入地回顾了三值量化。
- 详细比较了三值量化过程中的公式构建与优化。
- 深入探讨了不同三值量化方法之间的联系，这些联系也可推广至多比特量化。

本综述的整体结构如图1所示。第2节简要介绍三值量化。第3节阐述如何基于投影函数构建三值量化的公式。第4节从近端算子的角度总结各种优化方法。第5节和第6节分别提出三值量化潜在研究主题的建议，并给出结论。

2 三值神经网络

神经网络层的一般表示形式为：

$$\mathbf{y} = \phi(\mathbf{w}^\top \mathbf{x}), \tag{1}$$

其中，$\mathbf{w} \in \mathbb{R}^{n\times m}$、$\mathbf{x} \in \mathbb{R}^n$ 是层的输入向量，ϕ 表示非线性激活函数，而$\mathbf{y} \in \mathbb{R}^m$ 是

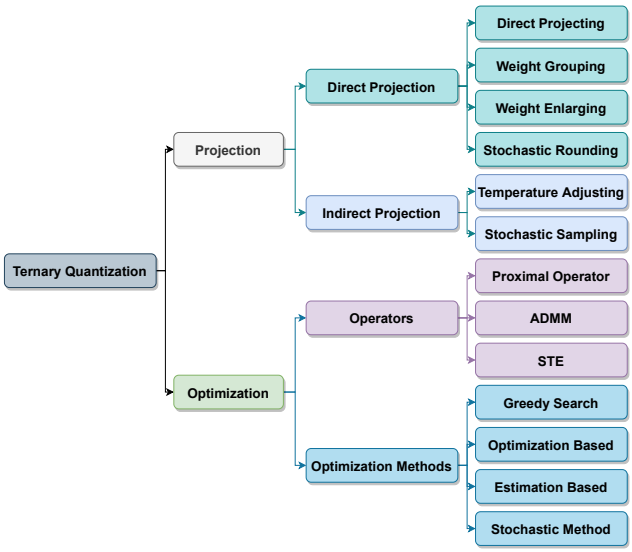

图1：这篇三值量化综述的整体结构。

输出特征向量。三值量化包含两个部分，即阈值化和投影：

$$\hat{w} = \mathcal{Q}(w) = \begin{cases} 1 : w > \Delta \\ 0 : |w| \leq \Delta \\ -1 : w < -\Delta, \end{cases} \tag{2}$$

$$s.t.\ \ \hat{w} \in \hat{\mathbf{w}}, w \in \mathbf{w}, \text{ and } \Delta \in \mathbb{R},$$

其中， Δ 表示量化阈值。优化目标可定义为：

$$\underset{\mathbf{w},\alpha}{\operatorname{argmin}} J(\mathbf{w}) = \|\mathbf{w} - \alpha\hat{\mathbf{w}}\|_2, \tag{3}$$

其中， α 表示缩放因子。给定 Δ， α 有一个闭式解 [32, 13]。更多详情请参见第4.1节。

2.1 简史

早期的三值量化研究旨在将模型部署到专用硬件上。最早的相关文献可追溯至1988年，见Tzi-Dar和M [50]的研究。由于硬件限制，权重必须量化为两个或三个离散值，才能将权重编程到超大规模集成电路芯片上。 [50] 的作者研究了若干种学习算法，以将连续权重映射为三值。该研究比较了六种方案，用于训练一个采用随机初始化的三层网络。他们没有说明具体的实现细节，也没有概述工作流程。根据他们的算法和综述，他们采用遗传算法结合梯度下降和模拟退火来解决三值权重优化问题。他们声称遗传算法优于其他方法。

1990 年，M. 等人 [35] 提出了详细的三值量化训练流程。与大多数现代研究不同的是，他们没有保留全精度权重的原始副本，而是在梯度更新后立即对模型进行量化。在随后的五年中，研究者们致力于扩展这项工作。然而，网络结构很简单，数据集也很小 [17, 14]。后来，Kyuyeon 和 Wonyong [30] 尝试采用一种类似于直通估计器（STE）[57]的方法进行量化。该网络规模更小，只有三层。

直到2016年，随着三值权重网络（TWN）的出现 [13], , 三值量化开始应用于大型模型（VGG [28], ResNet[27]）和大型数据集（ImageNet [22]）。因此，那些针对简单数据和模型设计的方法已不再适用。早期的 TWN 和训练式三值量化（TTQ） [8] 相对直观，其整体准确率与如今的三值量化方法有所不同。此后，基于 TWN 和 TTQ 的工作，涌现出若干与硬件相关的实现方案。此类方法侧重于硬件设计；网络也不再只有三层。

自2017年起，三值量化领域出现了一些新的分支。例如，三值约束优化 [10], 、面向低比特模型的知识蒸馏 [6],、矩阵分解 [41], 、分区 [4, 5, 42] , 、随机量化 [21, 9, 38] , 、平滑过渡 [40] 和加宽[7]。我们将在接下来的章节中详细阐述这些方法的发展。

2.2 三值优化概述

公式 (3) 中的缩放因子 α 和阈值 Δ 可以是不对称的。为了获得三值权重 $\{-\alpha, 0, \alpha\}$，大多数三值量化研究都着重于设计投影函数，以便能够部署到硬件设备上。然而，三值量化通常会带来三个问题：$\mathbf{w}$ 与 $\hat{}\mathbf{w}$ 之间的差异很大，模型可能陷入局部极小值 [2, 50], , 而投影会产生零梯度。因此，三值量化的目的之一是通过缩小 $\hat{}\mathbf{w}$ 与 $\mathbf{w}$ [13]之间的差距，提高量化模型的准确率。此外，由于当更新后的$\mathbf{w}$ 的幅值小于 Δ时，投影函数会抵消梯度更新，模型便会停滞在局部极小值处。使用 α 重新缩放$\mathbf{w}$ 有助于跳出这些局部极小值，从而提高量化模型的准确率[17]。因此， α 和 Δ 可以缩小 $\hat{}\mathbf{w}$ 与$\mathbf{w}$ [13], 之间的差距，并跳出局部极小值。

Tzi-Dar 和 M [50] 于1988年提出通过随机初始化跳出局部极小值。其缺点是无法将旧版本的权重用于后续训练。因此， [50] 进一步结合遗传算法（GA）， 通过混合停滞的权重来生成新的权重。M. 等人 [35] 的工作首次提出使用缩放因子来缓解这一局部极小值问题。 [35] 的作者将全精度权重投影到 {0, 1, -1}，然后使用 α 和 Δ 来调整量化权重的方向和幅度。他们的模型只有三层，每层 132 个单元。因此，他们使用贪心搜索来获得缩放因子 α 和阈值 Δ。另一个值得注意之处是， [35] 的工作仅在初始化时使用全精度权重。随后，直接在三值权重上更新梯度，这意味着全精度权重对后续优化并无帮助

化过程。

从 [30], 的工作开始，三值量化方法倾向于使用对偶平均次梯度算法 [61], ，也就是说，将梯度更新到全精度副本上，而不是三值权重上，这可以缓解由投影函数导致的局部极小值问题。通过这种方式，梯度可以在全精度权重上累积，直到**w** 的幅值大于 Δ。从另一个角度来看，全精度更新方法的直觉是，在最优全精度权重附近寻找最优三值权重。2014年之后的大多数工作在全精度权重上进行梯度更新；在使用大型数据集（例如 ImageNet）时，则使用预训练模型进行初始化。

2014年之后，越来越多的研究将公式 (3) 作为三值量化的基础公式，重点关注两个参数：一个是缩放因子 α，另一个是阈值Δ。直观地说，这两个参数的作用实际上是使**w** 和$\hat{}$**w** 更加接近。例如，Δ 和 α 会影响**w**的方向和幅度。模型权重先由 α 缩放，再由 Δ 投影。缩放因子 α 可以是固定值 [13] ，也可以是经过优化的目标 [8]。通常，阈值Δ 会将**w** 划分为零值和非零值。值得注意的是，Δ 通常通过估算而非优化得到。给定 Δ，α 存在闭式解（公式 (19) ）。此外，许多研究会迭代优化 α 和 Δ （见第 4 节）。给定估算得到的 Δ，首先通过公式 (19) 计算缩放因子（类似于 90 年代的方法），然后基于更新后的 α 再次估算 Δ。这一过程可以重复多次，直到**w** 和$\hat{}$**w** 之间的距离足够小。因此，我们得出结论：缩放因子α 和阈值 Δ 可以缩小 w 与 $\hat{}w$ 之间的距离，并跳出局部极小值。

虽然 α 和 Δ 有助于三值投影，但将连续变量投影到离散空间会引发优化问题，导致梯度不可微或为零。从近端优化的角度来看， [37],可以使用近端梯度下降近似求解这些问题。广泛使用的 STE 也可以视为近端优化的一种特殊情况 [37]。大多数优化过程都会保留一份模型权重的全精度副本，并基于量化产生的梯度进行优化。

本综述涵盖 2014年之后的三值量化研究，因为更早的研究基于小型网络。本综述的结构按照三值量化的投影函数和优化方法进行组织。

3 投影函数

投影函数的目的是生成离散权重值。不同于其他将投影函数分为确定性和随机性函数的研究，我们根据前向传播过程中权重的状态对其进行分类。本节接下来将总结直接投影函数和间接投影函数。（请注意，从本节开始，本综述中引用其他研究的方程将遵循原始研究中的定义，并注明这些方程的准确出处。）

3.1 投影函数概述

对于大多数基于投影的三值量化方法，反向传播梯度都是估计值。尽管 STE [57] 和 Hinton 的幻灯片最为常见，但类似方法早在 1994年就已在 [35]的研究中使用。与 STE 不同，在早期研究中，权重会在梯度更新后立即通过投影进行量化。显然，投影函数会抵消梯度更新，从而导致训练过程中陷入局部极小值 [50]的问题。直到最近，一些研究才表明，双重平均方法 [61]可以缓解权重抵消问题。在三值量化场景中，双重平均方法可以解释为：在反向传播过程中，使用投影后的（量化后的）权重值生成梯度，以更新全精度权重；如今，这已成为广泛采用的量化标准。

间接投影方法旨在通过采用渐进式投影过程来缓解梯度不准确的问题。一些研究采用逐元素投影，例如，混合三值和全精度值以创建混合值，并逐步调整三值的占比 [58] ，或使用基于温度的函数调整离散程度 [23, 9]。另一些研究则基于评分函数 [4] 或随机地 [3] 逐步混合三值和全精度值。除了渐进式投影方法外，随机方法还会根据多项分布采样离散权重，或向离散权重添加正态分布噪声以生成连续值 [38]。然而，对于深度神经网络而言，这些方法的采样效率可能是一个挑战。

3.2 直接投影

离散投影使用投影函数，在前向传播中生成不可微的离散值。它可分为完全离散方法和部分离散方法。部分离散方法包括逐组离散化和随机离散化。需要考虑的是，在反向传播过程中如何将梯度更新到权重上，以及如何求解相应的 α 和 Δ。三值量化的关键在于找到合适的缩放因子 α、阈值Δ和次梯度，以最小化$\hat{}$**w**与**w**之间的差异。

直接投影

最早的直接投影方法于1988年[50],被提出，但没有提供任何训练过程的细节。

首个详细的三元训练出现在1990年（ [35]的公式（5-8））：

$$\mathrm{E_h(A)} = \sum_{\mathrm{s}} \sum_{\mathrm{k}} \sum_{\mathrm{j=1}}^{\mathrm{N_{s-1}}} \left(\mathbf{w'}_{\mathrm{kj}}^{(s)} - \mathrm{A}^{-1} < \mathbf{w}_{\mathrm{kj}}^{(s)} \mathrm{A} > \right)^2 \quad (4)$$

$$\mathbf{w''}_{\mathrm{kj}}^{(s)} = \left\langle A_h \mathbf{w'}_{\mathrm{kj}}^{(s)} \right\rangle . \quad (5)$$

$$\mathbf{w''}_{\mathrm{kj}}^{(s)}(t+1) = \left\langle \mathbf{w''}_{\mathrm{kj}}^{(s)}(t) + \eta \Delta_k^{(s)} \right\rangle \quad (6)$$

其中， A 是缩放因子， $\langle \cdot \rangle$ 表示舍入， Δ 是梯度。我们可以看到， [35] 的工作旨在找到一个

合适的缩放因子 A，使缩放后的$\hat{}\mathbf{w}$ 与$\mathbf{w}$之间的距离最小。然后使用缩放后的$\hat{}\mathbf{w}$ 进行前向传播。他们的方法与直通估计器（STE）的区别在于，公式（6）在梯度更新后直接对$\mathbf{w}$ 进行舍入。

Kyuyeon 和 Wonyong [30]在前向传播中将$\mathbf{w}$ 投影到离散空间以生成梯度，并保留一份权重值的全精度副本用于梯度更新。该方法还将工作扩展到更高位量化的情形（[30]中的图 3 和公式 (5-8)）：

$$\begin{aligned} w_{ij,new} &= w_{ij} - \alpha \left\langle \tfrac{\partial E}{\partial w_{ij}} \right\rangle \\ w^{(q)}_{ij,new} &= Q_{ij}\left(w_{ij,new}\right) \\ Q(x) &= \Delta \operatorname{sgn}(x) \cdot \min\left\{ \left\lfloor \tfrac{|x|}{\Delta} + 0.5 \right\rfloor, \tfrac{M-1}{2} \right\} \end{aligned} \tag{7}$$

其中，$\frac{M-1}{2}$ 表示位数，Δ 以贪心方式进行优化，而 $\langle\cdot\rangle$ 返回均值。梯度 $\frac{\partial E}{\partial w_{ij}}$用于更新全精度$\mathbf{w}$，这实际上是对直通估计器的使用。Wonyong 等人 [52] 使用相同的量化方法，从模型复杂度的角度研究浮点权重与三值权重之间的性能差距。

TWN 的作者 Fengfu 等人 [13], 在后续工作中使用直通估计器进行量化。在他们的工作中，阈值 Δ 是一个估计值，而缩放因子 α 有一个闭式解（公式 (19)）。Shen [46] 将 TWN 应用于使用二值权重值的激活量化。Yin 和 Xin [55] 引入两个缩放因子，并使用一个估计的 Δ 来获得更准确的三值。Yin 等人 [56] 将缩放因子 α 限制为二的幂形式，并给出 α的下界；此外，它也像其他工作[8, 13] 一样估计 Δ。 [18, 31] 的工作使用贪心二分搜索来寻找最优的 Δ，并定义了一个评分函数来衡量$\mathbf{w}$ 与 $\hat{}\mathbf{w}$ 之间的差异（[18]中的公式 (3)）。Wu [53] 提出一种基于 k 均值的预处理方法，将每个权重值分配给聚类中心，即 $\pm\alpha$，0。缩放因子 $\pm\alpha$ 起到阈值 Δ的作用。作者使用一种类似于 TWN 的方法来获得 α。最终量化通过标准直通估计器完成。

权重分组

分组方法逐渐增加投影后的全精度权重（即 Q($\mathbf{w}$)）所占的比例。分组的优势在于，与完全量化的权重相比，使用混合精度权重可以带来更小的误差，从而产生更准确的梯度。误差可通过以下方式进行测量：

$$\begin{aligned} \|\mathbf{y} - \mathbf{x}^\top \mathbf{w}\|_2^2 &\le \|\mathbf{y} - \mathbf{x}^\top (r\mathbf{w} + (1-r)\mathbb{Q}(\mathbf{w}))\|_2^2 \\ &\le \|\mathbf{y} - \mathbf{x}^\top \mathbb{Q}(\mathbf{w})\|_2^2, \end{aligned} \tag{8}$$

其中，r 控制 $\mathbf{w}$ 的投影比例。其中一种极端情况见于 [64]:

$$\begin{aligned} \min_{w_j} L = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} \max\left(0, 1 - y_i \alpha \left(\mathbf{w}_{\neg j}^\top \mathbf{z}_i + w_j z_{ij}\right)\right) \\ + \lambda\alpha^2 w_j^2 + \lambda\alpha^2 \mathbf{w}_{\neg j}^\top \mathbf{w}_{\neg j}, \end{aligned}$$

$$\text{s.t.} \quad w_j \in \{-1, 0, 1\} \tag{9}$$

其中，$\mathbf{z}$ 是输入，L 表示合页损失，而$\mathbf{w}_{\neg j}$ 表示全精度权重向量，其第 j个元素等于零。公式 (9) 表明：可以逐元素执行三值量化，以获得最优的三值权重。

或者也可以反过来进行；例如，许多其他方法 [4, 54, 42] 固定三值权重，但更新全精度权重。在训练过程中，固定的三值权重部分逐渐增加。Aojun等人 [4] 以权重幅值作为固定三值权重的参考。Yiming等人 [54] 则采用按列优化，即固定其他列来优化某一列。Peisong等人 [42] 和 Jiwei等人 [23] 分别执行逐元素量化和逐层量化。

此外，K. *et al.* [26] 在 $|\pm 0.5|$ 周围引入了一个全精度缓冲区，并使用 h 控制该区域的宽度：

$$w(t) = \begin{cases} -1 & \text{if } w_h(t) \le -0.5 - h, \\ 0 & \text{if } w_h(t) \ge -0.5 + h \vee w_h(t) \le 0.5 - h, \\ 1 & \text{if } w_h(t) \ge 0.5 + h, \\ w(t-1) & \text{otherwise} \end{cases} \tag{10}$$

通过设置连续区域 $[\pm 0.5, \pm h]$，一旦区域内的权重值获得足够的梯度，它们就有机会跳出该区域并变为离散值。然而，这种设计无法保证所有权重都能跳出。Dubey [12]将分组概念与 TWN 的量化方案相结合。更具体地说，将 $\mathbf{w}$ 划分为 N个正交子集，并通过暴力搜索优化最优的 α 和 Δ（最小化$\mathbf{w}$ 与$\hat{}\mathbf{w}$之间的 $L2$ 距离）。Dubey [12] 的作者还发现，通过输入通道进行分组可以获得更好的三值量化结果。SYQ（[24]）提出了与 [12] 类似的方法，但采用了不同的分组方案，并使用随机梯度下降 [8]优化组级 α。Peisong 和 Jian [41] 以贪心方式将全精度权重矩阵分解为三个子集，并对这些子集执行三值量化；然而，训练细节并不明确。INQ（[4]）对幅值较大的全精度权重进行量化并冻结。其余可调整的权重值则进一步优化并逐渐量化。Aojun *et al.* [5]的工作定义了三值中心，然后逐渐增加三值中心周围的量化比例。Lukas 和 Luca [34]将权重随机划分为三值组和全精度组。首先，冻结三值部分，并更新全精度部分。然后冻结全精度部分，并将三值部分更新为全精度。重复这一过程，同时增加$\hat{}\mathbf{w}$ 所占的比例，直到量化完成。

权重扩展

复制层权重 [62] 并增加层宽度[7] 可以提升模型的容量。我们将这种操作称为扩增。

Asit 等人 [7] 研究了增加模型冗余（例如增加层宽度和三值滤波器数量）对三值量化准确率的影响。Zhezhi等人 [62] 研究了添加额外的三值残差层以提升模型性能。每一层都包含多份权重副本，这些副本在经验上被赋予不同的 Δ。Weixiang 等人 [51] 通过将两个带有缩放因子 α的二值权重相加来生成三值，从而避免了 Δ的不确定性：$\{\alpha, -\alpha\} + \{\alpha, -\alpha\} = \{2\alpha, 0, -2\alpha\}$。类似地，Yue 等人 [59] 将两个带系数的三值权重叠加，以生成新的三值。

Hassan等人 [19] 提出了一种可分离三值结构。训练过程中，将两个三值权重叠加为混合权重。混合权重可拆分为两个三值权重，以实现快速推理。推理时，可以分别计算一层的两个内积，但计算开销会增加一倍。Hyungjun 等人[20] 通过调整 BatchNorm层的偏置，将三值激活解耦为两个二值激活。三值激活可分解为两个二值激活之和。

总而言之，提出扩增是为了弥补三值量化导致的模型容量降低。采用扩增后，性能可以得到提升，但也会带来额外的计算开销。

随机舍入

上述方法通常被归类为确定性方法。对于随机方法，Zhouhan 等人 [63], 提出了较早的随机三值量化研究，该研究是二值量化 [36]的扩展工作。A.等人 [1] 和 Hyungjun等人 [20] 的研究表明，随机化舍入或随机舍入可以提供无偏离散投影。Zhouhan 等人 [63]在前向传播中应用随机三值量化，并对输入应用二次幂量化，以消除反向传播过程中的乘法运算。Hande 等人[18] 使用预训练的全精度（教师网络）模型来指导具有相同结构的三值（学生网络）模型。该方法对教师网络的激活值进行随机量化，然后通过逐层算法使学生网络模仿这些激活值。激活值在权重之前进行量化。

与常用方法在离散空间中获取梯度、并在连续空间中更新权重不同，在 Li [32], 的工作中，梯度会以随机方式投影到离散空间，以更新三值权重。权重值总共有六种可能的转变，即$\{-1\} \rightarrow \{0,1\}$、$\{0\} \rightarrow \{-1,+1\}$和$\{1\} \rightarrow \{-1,0\}$（见 [32]中的图 3）。投影概率取决于梯度的幅值和方向。

总之，随机舍入会根据不同的度量（例如权重幅值），以一定概率选择两个最近网格点中的一个。尽管随机舍入可以减少投影中有偏的梯度，但采样速度较慢 [参见]。因此，生成权重的效率很低，尤其是在权重元素数量较大时，这会降低模型的训练和推理效率。

3.3 间接投影

上述直接投影方法存在不可微性问题。间接投影方法尝试部分或逐步绕过这种不可微性。换言之，它们尽可能在连续空间中训练量化模型。间接投影方法可分为渐进式和随机离散投影方法。

温度调节

为了逐步调整离散程度，一种常见的方法是在投影函数中引入温度参数，例如，双曲正切函数 [44], Sigmoid函数 [23], 和 GumbelSoftmax [9]。温度越高，函数就越接近阶跃函数。温度参数控制函数的陡峭程度。调整温度可以在量化过程中调节离散程度。

在 Jiwei等人 [23] 和 Hassan等人 [19], 的研究中，连续权重经过Sigmoid函数，随着Sigmoid函数的温度逐渐升高而变为离散值：

$$\sigma(Tx) = \frac{1}{1+\exp(-Tx)}. \tag{11}$$

提高温度 T 可以使权重值逐步从连续值转变为离散值。Peiqi等人[40] 将 TTQ 的研究扩展到循环神经网络，并使用斜率因子调节激活的陡峭程度以进行量化。不同于 TWN 直接使用指示函数，Yu等人 [58] 提出了软阈值函数来实现量化。具体而言，他们提出了 $L1$ 和 $L2$ 正则化项，以调节量化程度。随后，逐步增大的 λ 用于调节正则化项的强度，从而将全精度权重转换为离散值。因此，ProxQuant [58] 可视为一种渐进式量化方法。基于 GumbelSoftmax的方法，如 [9], ，也通过逐渐改变温度参数，将连续权重转换为离散值。

随机采样

随机量化方法有两种。一种是将连续权重随机投影到离散值（第3.2节）。另一种随机量化方法则通过添加噪声，使离散权重重新变为连续权重，以便于优化 [21], ；或者逐步将随机投影从连续空间转换到离散空间 [9]。在 Oran等人 [38], 的工作中，多项分布通过两个 Sigmoid函数实现，层权重用作概率参数，但代价是权重大小翻倍。对多项分布进行采样后，计算$\hat{}\mathbf{w}$ 的均值和方差。在得到均值和方差后，前向传播中会使用噪声$\mathcal{N}(0,\ I)$来生成连续权重。推理时，通过直接从多项分布中采样来获得离散权重（更多详情请参阅第4.2节）。Christos等人 [9]使用 Gumbel-Softmax 和温度参数。由于它像基于 Sigmoid函数的方法一样采用逐渐变化的温度，因此可以将其视为 LR-net [38], 的增强版本。权重的最终形式是离散值，而 LR-net 在推理时直接采样离散权重，效率较低。随机采样可以采样多个权重以构成集成模型，但整体准确率还有提升空间。

总之，间接投影存在运行效率低的问题。温度调节方法使用 Sigmoid函数将连续权重逐渐转换为离散值。其准确率较高，但训练耗时较长。例如， [23]的量化方法需要进行耗时的逐层量化，并分为三个阶段：权重量化训练、使用量化权重进行激活量化训练，以及使用

量化激活值和权重共同进行微调。训练时间取决于网络的深度。随机方法受制于低效的采样——网络参数越多，采样速度就越慢。

4 优化方法

三值量化的优化方法有多种分类方式，例如基于优化、基于估计和基于贪心搜索。还可以采用替代训练方案、知识蒸馏或额外的损失项来辅助优化过程。我们总结了各种形式之间的联系，并利用近端映射统一直接投影优化方法。

近端算子

由于近端算子可视为广义投影， [37], 我们使用近端算子来统一三值量化方法。在本综述中，三值设定下的近端算子定义如下：

$$\operatorname{prox}_{I_{\mathbb{Q}}}(w)=\underset{x\in\mathbb{R},\alpha\in\mathbb{R}}{\operatorname{argmin}}\left(I_{\mathbb{Q}}(x)+\frac{1}{2\lambda}\|w-\alpha x\|_2^2\right), \tag{12}$$

其中 α 是一个缩放因子。如果 $I_{\mathbb{Q}}(x)$ 是指示函数：

$$I_{\mathbb{Q}}(x)=\begin{cases}0 & x\in\mathbb{Q}(x),\\ +\infty & x\notin\mathbb{Q}(x),\end{cases} \tag{13}$$

那么 $I_{\mathbb{Q}}$ 的近端算子可化简为欧几里得投影 $P(\cdot)$ 到闭非空凸集 $\mathbb{Q}(x)$ 上：

$$P_{\mathbb{Q}}(w)=\underset{\alpha,x\in\mathbb{Q}(x)}{\operatorname{argmin}}\|w-\alpha x\|_2. \tag{14}$$

公式 (14) 等价于公式 (3)。三值量化的优化过程可以看作欧几里得投影，即将全精度权重投影到离散集合 $\mathbb{Q}(\cdot)$ 中。

交替方向乘子法

交替最小化常用于对两个变量进行优化。三值量化可以看作约束凸优化，即交替方向乘子法 (ADMM) [48] 的一种特殊情况。我们以 ADMM 为一般示例，总结三值量化中广泛使用的交替训练方案。考虑一个一般的约束凸优化问题：

$$\begin{array}{ll}\text{minimize} & f(w)\\ \text{s.t.} & w\in\mathbb{Q}(w).\end{array} \tag{15}$$

上述问题的 ADMM 形式为：

$$\begin{array}{ll}\text{minimize} & f(w)+I_{\mathbb{Q}}(z)\\ \text{s.t.} & w-z=0.\end{array} \tag{16}$$

训练算法如下：

$$\begin{aligned}w^{k+1}&:=\operatorname{prox}_f(w)=\underset{w}{\operatorname{argmin}}\left(f(w)+\frac{1}{2\lambda}\left\|w-z^k+u^k\right\|_2^2\right),\\ z^{k+1}&:=\operatorname{prox}_{I_{\mathbb{Q}}}(w)=P_{\mathbb{Q}}\left(w^{k+1}+u^k\right),\\ u^{k+1}&:=u^k+w^{k+1}-z^{k+1},\end{aligned} \tag{17}$$

其中 k 表示训练迭代。交替方向乘子法在 f 和 $I_{\mathbb{Q}}$ 的近端算子可以高效计算时最为有用，但 $f+I_{\mathbb{Q}}$ 的近端算子不易计算（参见《Proximal Algorithms》[37]第 154 页）。因此，可以轻松地使用交替最小化方法执行量化 [30, 42, 64, 23, 54, 33, 58, 41]。例如，给定特定的 Δ 和 α，先优化 $\operatorname{prox}_f(w)$，再优化 $\operatorname{prox}_{I_{\mathbb{Q}}}$，以获得更新后的 Δ 和 α，重复此过程直至模型收敛。

直通估计器

使用直通估计器的三值量化优化过程等价于采用对偶平均的三值投影函数 $P_{\mathbb{Q}}(w)$ [61]:

$$\begin{cases}\hat{w}_t:=P_{\mathbb{Q}}(w)=\underset{\alpha,x\in\mathbb{Q}(x)}{\operatorname{argmin}}\|w-\alpha x\|_2,\\ w_{t+1}=w_t-\eta_t\nabla L\left(\hat{w}_t,\right)\end{cases} \tag{18}$$

其中， t 表示训练迭代， η_t 表示学习率。当=1时，公式 (14) 就变成了非常常见的量化函数，例如 [1, 16, 34, 32, 29, 63, 26, 18, 30]。许多硬件量化文章 [43, 1, 47, 16] 也倾向于采用这种方法。然而，投影到 $\{\pm1,0\}$会增大$\mathbf{w}$ 与$\hat{}\mathbf{w}$之间的距离。

考虑 $\alpha>0$的情况，著名论文 TWN [13] 提出使用公式 (14)。除了上述投影 P_w之外，TWN 的工作还使用近似的 Δ 来最小化优化目标，然后将$\mathbf{w}$ 投影到 $\alpha\hat{}\mathbf{w}$上。与 TWN 的工作不同，TTQ[8] 将 α 作为优化参数。

4.1 直接投影

直接投影方法通常包括 α 和 Δ。在下文中，将从如何获得 α 和 Δ这三个方面概述相关工作：基于估计、基于优化和基于贪心搜索的方法。

基于估计的方法

基于估计的方法通常以公式 (14) 为基础。如公式 (14) 所示，估计 α 和 Δ 的目的是找到 $\mathbf{w}$与 $\hat{}\mathbf{w}$之间最短的 $L2$ 距离。给定固定的 Δ， α 的闭式解为：

$$\alpha=\frac{1}{\|\mathbf{w}\|_0}\sum_{i=0}^{n}|w_i|. \tag{19}$$

公式 (19) 的矩阵形式可见于 Cong 等人 [10]的研究。基于估计的方法的一个缺点是，Δ没有显式解，这会导致 α 不准确。因此，TWN 通过 $\frac{0.7}{n}\sum_{i=1}^{n}|w_i|$直接估计 Δ，而 TTQ 使用 $\Delta=0.05\times\max(|\mathbf{w}|)$。如上所述，在公式 (14) 中，当 α 估计为 1 时，量化操作等价于将数值投影到 $\{0,1,-1\}$ [30, 45]。

尽管许多研究并未提及公式 (14) 的形式，但根据近端映射的定义，任何采用投影函数的方法都可归入这一基于估计的类别。例如，在 ELBNN [10], 中，作者通过增广拉格朗日扩展了近端算子，即采用交替方向乘子法进行复杂投影。ELBNN 定义了 α的矩阵形式：

$$\alpha=\frac{\mathbf{w}^\top\hat{\mathbf{w}}}{\hat{\mathbf{w}}^\top\hat{\mathbf{w}}}=\frac{1}{\|\hat{\mathbf{w}}\|_0}\sum_{i=0}^{n}|w_i|, \tag{20}$$

这与公式 (19) 相同。

此外，ProxQuant [58] 在公式 (14) 中采用投影函数$\alpha = 1$，与 TWN 的做法相同。ProxQuant 的作者提出了两种正则化项，并将量化函数定义为：

$$\hat{\mathbf{w}} = \begin{cases} \textsc{SoftThreshold}(\mathbf{w}, \hat{\mathbf{w}}, \lambda), (L1\ term), \\ \frac{\lambda\hat{\mathbf{w}}+\mathbf{w}}{1+\lambda}, (L2\ term), \end{cases} \tag{21}$$

其中，λ 控制量化权重$\hat{}\mathbf{w}$所占的比例。Peisong 和 Jian [41] 提到，缩放因子 α 应与行数和列数之和的平方根成正比，但没有详细说明所提出的估算方法。Aojun 等 [4] 使用 2^n 作为缩放因子，并逐步进行分组量化。Shen [46] 使用$\Delta = \frac{0.4}{n}\sum_{i=1}^{n}|w_i|$ 生成三值激活。Aojun 等 [5] 添加了 $L1$ 和 $L2$ 正则化项，以促使权重趋近于三值。α 通过 $\text{mean(w)} + 0.05\text{max(w)}$ 估算。训练过程较为复杂：首先使用收敛后的全精度权重得到 α，然后进行分组量化。接着，得到新的 α，并逐组量化更多权重。Zhezhi 等 [62] 使用 $\alpha = \text{mean(w)}$（TWN）和$\Delta = 0.05 \times \text{max(|w|)}$（TTQ）作为量化设置。此外，他们还引入了一个带有独立 α的额外残差层。Weixiang 等 [51] 通过添加两个二值权重得到三值权重，并给出了 α的闭式解。 [51] 这项工作的一个优势是，不再需要 Δ。 [51] 的作者还提出了精细化梯度（ [51]中的公式 (13)）；然而，他们发布的代码中使用了知识蒸馏。因此，很难知道不使用知识蒸馏时的结果。

对于基于估计的方法，Δ 控制$\mathbf{w}$中的零值数量。给定一个 Δ，α 可以得到闭式解。然而，由于训练过程中学习率和权重衰减会持续产生大量近零值，因此很难明确确定 Δ。因此，如果估计的 Δ 不准确，α 也会不准确。优化 α和 Δ时，可以将适当的稀疏度作为考量因素。

基于优化的方法

为解决上述 α 和 Δ 不准确的问题，TTQ [8] 提出了两个通过优化得到的缩放因子：

$$\frac{\partial J}{\partial \alpha^{\pm}} = \sum_{i=1}^{n^{\pm}} \frac{\partial J}{\partial w_i^{\pm}}, \tag{22}$$

其中“±”表示正值或负值部分。例如，α^- 表示负权重值的缩放因子w^-，并通过 $\sum_{i=1}^{n^-}\frac{\partial J}{\partial w_i^-}$ 更新；n^- 表示$\mathbf{w}$中的负值数量。对于 TTQ 中的 Δ，其形式与 TWN 类似，即 $\Delta = 0.05 \times \text{max}(\|\mathbf{w}\|_1)$，这表明 $\|\mathbf{w}\|_1$ 与层稀疏度有关。因此，TTQ 的作者研究了模型稀疏度对三值量化的影响，并得出结论：“稀疏度超过 *50%* 后，模型容量会过度下降，导致误差增大。稀疏度在 *30%* 到 *50%* 之间时，误差最小。” [8]。Asit等人 [7] 采用与 TTQ 相同的方法，但通过加宽网络（WPRN）进行扩展。Peng等人 [43] 提出了一种可学习的 Δ，方法是将其替换为$\frac{\alpha}{2}$。Julian等人 [24] 对权重值进行分组，并为每组分配缩放因子。随后，像 TTQ 一样，使用随机梯度下降优化这些细粒度缩放因子。Jiwei等人 [23]使用 Sigmoid函数和温度系数 T对量化进行公式化表示：

$$\mathbb{Q}(w) = \frac{1}{1 + exp(T \times (\beta w + b))}, \tag{23}$$

其中，β 和 b 是量化参数 w的可学习系数。通过这种方式，可使用随机梯度下降优化每个参数（ [23]的公式（8–11））。Yuhang 等人 [60] 采用了与 [23] 相同的可学习系数，方法如下：$\bar{w} = \beta w + b$，其中β 和 b 均通过直通估计器学习，然后执行三值量化。然而，[60] 并未像 [23]那样提供详细的梯度更新规则。Yue 等人 [59]将三值$\hat{}\mathbf{w}$ 与$\mathbf{w}$ 之间的残差值叠加，得到一个三值投影函数，其输出为 $\{-2\alpha, 0, 2\alpha\}$，且阈值化时无需 Δ。此外，该方法改进了 α 的梯度，并且很容易扩展到任意位数（参见 [59]的公式（13–16 和 19））。

Ryan 等人 [45] 通过加入损失正则化项，将二值量化与三值量化相结合，从而扩展了 TTQ 的工作：

$$\min\left(||w| + \alpha|^p, ||w| - \alpha|^p, \tan(\beta)|w|^p\right) \tag{24}$$

通过将 β 从 $\frac{\pi}{4}$ 调整到 $\frac{\pi}{2}$，他们可以控制三值量化的强度（见 [45]中的图 2）。然而，p 在实验中被设为 1，且未提供有关优化 β 和 α的细节。Yiming 等人 [54]通过在目标函数中加入正则化项，使全精度权重更接近聚类中心（三值），随后使用直通估计器进行三值量化。Lu和T. [33]将二阶信息（海森矩阵）项加入目标函数，以调整梯度更新。

基于贪心搜索的方法

贪心搜索方法是获得 α的另一种途径。 [35, 17, 18] 的研究采用贪心搜索方法，在指定范围内确定缩放因子 α；而 Kyuyeon 和 Wonyong [30]则不设范围，直接搜索 α。Hande等人 [18]和Leroy [31]使用二分搜索算法来寻找最佳 $\pm\alpha$。Wu [53] 使用 k 均值，在每次梯度更新后持续更新$\mathbf{w}$ 的三值中心。$\mathbf{w}$ 的取值会更新为其对应的中心。随后，为下一次训练迭代计算$\mathbf{w}$ 的新中心。为提高对 TWN 中 α 和 Δ的估计准确性，Dubey [12] 使用两组 α 和 Δ来完成量化，并通过穷举搜索求得它们。具体而言，每个输入通道都有一个独立的缩放因子 α。Yin 等人 [56]使用二的幂缩放因子进一步提升性能。当量化位数大于 2 时，Δ 采用 $0.75 \times max(||w||_1)$ 的固定形式，这与 TTQ 中估算的 Δ 类似。Lu和T. [33] 使用牛顿近端法来促进三值量化；计算 α时，将二阶信息$\mathbf{d}$用作缩放因子：

$$\alpha = \frac{\|\hat{\mathbf{w}} \odot \mathbf{d} \odot \mathbf{w}\|_1}{\|\hat{\mathbf{w}} \odot \mathbf{d}\|_1}. \tag{25}$$

定义贪心搜索函数 find(·)，以帮助获得 α。将 Δ 赋值为 $\frac{a}{2}$（ [33]中的推论 3.1）。Peisong 等人

[42] 使用贪心搜索来寻找优化后的α。具体而言，在每次迭代中，它只更新$\hat{}\mathbf{w}$中的一个元素，而固定其他所有元素。它只需检查 $n \times 2^k$ 种可能性，并且位数 k 通常很小。Yiming等人 [54] 最小化公式 (3) 的一个变体，以促使权重值接近 $\{0, \pm\alpha\}$:

$$\underset{\alpha, z \in \{0,1\}}{\operatorname{argmin}} \left\| \mathbf{w} - \alpha \hat{\mathbf{w}}^T \mathbf{z} \right\|^2, \tag{26}$$

$$\alpha = \frac{\mathbf{w}\mathbf{z}^\top \hat{\mathbf{w}}}{\hat{\mathbf{w}}^\top \mathbf{z}\mathbf{z}^\top \hat{\mathbf{w}}}, \tag{27}$$

其中，$\mathbf{z}$ 表示类似剪枝掩码的稀疏度指示向量。 [54] 的作者定义了一个贪心函数 H(·) 来求解 $\mathbf{z}$，即找到合适的 $\mathbf{w}$ 稀疏度，以获得最优的α。Lukas 和 Luca [34] 采用逐组随机量化，并逐步增加量化部分。在初始化阶段，通过α^w 对预训练权重进行重新缩放，以缩小$\mathbf{w}$ 与$\hat{}\mathbf{w}$ 之间的 $L2$ 距离（[34]中的公式 (5)）。至于如何找到优化后的 α，他们在 $[0, \max(\|\mathbf{w}\|_1)]$范围内均匀选取 1,000 个点进行网格搜索。Zi-jian 和 Liang [64] 展示了一种经典的近端映射量化方法。他们将支持向量机模型量化为三值。优化过程不涉及梯度下降（公式 (9)）。相反，该方法穷举每个 w 对应的 $\{-1, 0, 1\}$ 所有可能取值，并选择使损失最小的取值。

总而言之，尽管直接投影直观且便捷，但离散权重会导致梯度不准确。因此，基于近端算子和直通估计器，许多工作提出了不同的改进思路，以缓解梯度不准确的问题，或获得更准确的 α 和 Δ。然而，目前尚无详细分析阐明 α、 Δ与$\mathbf{w}$之间的联系。此外，近年来分类准确率并未取得显著提升。

4.2 间接投影

间接投影可以抵消直接投影的缺点，这一观点基于 Matthieu 等人[36]的研究： *[25]* 和 *[49]* 表明，随机舍入或随机量化可以提供无偏的离散投影。

在 Zhouhan等人 [63], 的研究中，他们在前向传播中采用随机方法，并提出约束函数$\bar{w}$ = 硬 Sigmoid(w) [36] 以确保 $\bar{w} \in \{0, 1\}$。实际上，$\bar{w}$ 是随机值生成器的一个参数。随机量化过程定义如下:

$$\begin{array}{l} P(\hat{w} = 1) = \bar{w}; P(\hat{w} = 0) = 1 - \bar{w}; \text{if } \bar{w} > 0 \\ P(\hat{w} = -1) = -\bar{w}; P(\hat{w} = 0) = 1 + \bar{w}; \text{if } \bar{w} \leq 0 \end{array} \tag{28}$$

然而，随机量化只能缓解梯度不准确的问题，而且在量化较大模型时，采样速度较慢。尽管[63] 的实验结果优于确定性方法，但几乎所有后续确定性方法的表现都超过了 [63]的工作。此外， [63]的作者还展示了在反向传播过程中将梯度量化为 3 至 4 位的可能性。与其他直接更新权重全精度副本的工作不同，GXNOR-Net [32] 将梯度幅值映射为转移概率。随后，随机三值量化由这些概率控制。

Oran 等人等人 [38] 的另一项类似工作将 θ 作为多项分布 M() 的参数，以生成离散权重值 $\hat{}w = \text{M}(\theta)$。为了使离散值 $\hat{}w$重新变为连续值，作者采用了以下过程:

$$\begin{array}{l} \mu_{ij} = \mathbb{E}_{\theta_{ij}}\left[\hat{w}_{ij}\right] \text{ and } \sigma_{ij}^2 = \text{Var}_{\theta_{ij}}\left[\hat{w}_{ij}\right], \\ m_i = \sum_j \mu_{ij} h_j \text{ and } v_i^2 = \sum_j \sigma_{ij}^2 h_j^2 \\ \epsilon \sim \mathcal{N}(0, I), \\ w = m + v \odot \epsilon, \end{array} \tag{29}$$

其中， h 表示层的输入（ [38]中的公式 (4)）。换言之，他们利用正态分布噪声 ϵ、均值 m_i和方差 v_i^2，通过重参数化技巧 [39] 生成连续权重。这样，梯度便能通过投影操作。

知识蒸馏（KD）也可用于提升三值量化的性能。Geoffrey等人 [15]提出了知识蒸馏方法，将知识从较大的神经网络（教师网络）迁移到较小的神经网络（学生网络）。Asit 和 Debbie [6]提出将 KD 应用于量化；在他们的研究中，他们比较了不同的场景，例如同时训练教师网络和学生网络，或使用预训练的教师网络训练学生网络。除了比较结构相同的教师网络和学生网络外，他们还比较了结构不同的场景。结果表明，KD 能显著提升量化性能，但在不同网络结构和方案下应用 KD 并不会带来额外提升。

5 机遇与未来工作

三值量化面临的问题是训练效率低和准确率不高。大多数使用大型数据集的三值量化方法都基于预训练模型进行训练，因此耗时更长。准确率较低的原因在于使用估计梯度更新层权重。即使增加位宽，直通估计器中的估计偏差仍会导致准确率损失，因为在执行反向传播时，连续空间至关重要。此外，常用的加速仅适用于 16 位或 8 位权重，而三值推理的硬件实现尚未得到广泛应用。除此之外，只有少数研究（如 TTQ 等）探讨了模型稀疏度与量化之间的关系。从码本的角度来看，矢量量化（VQ）与三值量化相似。三值量化可以被视为固定码本的一种特殊情况，在这一领域还可以开展更多研究。三值量化主要体现在对内存占用的优化上，而 VQ 的优势主要体现在对硬盘空间的优化上。将二者结合，可以同时优化硬盘空间和内存占用。还可以从随机视角和生成模型视角出发，设计出更多三值量化方法。

另一个问题是，大多数研究都使用全精度权重作为初始化，以获得三值权重。直观来看，这可能需要修正。Jan 等 [21]提出了一种受约束全精度模型的训练流程，无需进行量化感知训练，即可获得非常准确的量化模型。Dan 等 [11]指出，找到三值权重附近的最佳全精度权重，然后应用

常规三值量化可以提升量化性能。换言之，当前的初始化方法可能需要调整，以提高准确率。此外，未来可能会出现通用的量化方法。

6 结论

本文从投影函数和优化函数的角度总结了三值量化。我们首先简要介绍三值量化的发展史，例如，首种三值量化方法的起源，以及 STE 引入之前的三值量化方法。我们介绍了投影函数，包括直接投影、权重分组、扩展、随机舍入、温度调节和随机采样。我们的分类提供了一种有别于传统分类的新视角，例如将量化方法分为均匀、非均匀、对称和非对称量化方法。此外，我们的分类也适用于高比特量化研究。我们通过近端优化揭示了各种三值量化方法之间的内在联系。我们说明了交替方向乘子法（ADMM）如何解释三值量化中广泛采用的交替训练方案。我们还探讨了三值量化中 STE、对偶平均和近端算子之间的联系。本文总结了当前三值量化领域的研究，并为高比特量化研究提供了见解。

参考文献

[1] Laborieux A ., Bocquet M ., Hirtzlin T ., Klein J .-O., Diez L . Her- rera, Nowak E., Vianello E., Portal J.-M. 和 Querlioz D. 基于阻变随机存取存储器突触的三值神经网络低功耗内存实现。载于*2020*年第二届 *IEEE* 人工智能电路与系统国际会议（*AICAS*），第 136–140 页，2020年8月。

[2] Gholami Amir、Kim Sehoon、Dong Zhen、Yao Zhewei、Ma- honey Michael W 和 Keutzer Kurt。《高效神经网络推理量化方法综述》，2021年。[3] Fan Angela、Stock Pierre、Graham Benjamin、Grave Edouard、Gribonval R´emi、Jegou Herve 和 Joulin Armand。《通过量化噪声训练实现极致模型压缩》，2020年。[4] Zhou Aojun、Yao Anbang、Guo Yiwen、Xu Lin 和 Chen Yurong。《增量网络量化：迈向采用低精度权重的无损卷积神经网络》，2017年8月。[5] Zhou Aojun、Yao Anbang、Wang Kuan 和 Chen Yurong。《面向低比特深度神经网络的显式损失误差感知量化》，第9426–9435页，2018年。[6] Mishra Asit 和 Marr Debbie。《Apprentice：使用知识蒸馏技术提升低精度网络的准确率》，2017年11月。[7] Mishra Asit、Nurvitadhi Eriko、Cook Jeffrey J. 和 Marr Deb- bie。《WRPN：宽化低精度网络》，2017年9月。[8] Zhu Chenzhuo、Han Song、Mao Huizi 和 Dally William J.《训练式三值量化》。载于*arXiv:1612.01064 [cs]*, 2016年12月。[9] Louizos Christos、Reisser Matthias、Blankevoort Tijmen、Gavves Efstratios 和 Welling Max。《离散化神经网络的松弛量化》。载于*arXiv:1810.01875 [csstat]*, 2018年10月。[10] Leng Cong、Li Hao、Zhu Shenghuo 和 Jin Rong。《超低比特神经网络：借助交替方向乘子法榨取最后一比特》，2017年9月。[11] Liu Dan、Chen Xi、Ma Chen 和 Liu Xue。《超球面量化：迈向更小、更准确的模型》。载于*IEEE/CVF*冬季计算机视觉应用会议论文集，第5262–5272页，2023年。[12] Naveen Mellempudi Abhisek Kundu Dheevatsa Mudigere Di- pankar Das Bharat Kaul Pradeep Dubey。《采用细粒度量化的三值神经网络》，2017年5月。[13] Li Fengfu、Zhang Bo 和 Liu Bin。《三值权重网络》，2016年11月。[14] Hoskins B. G.、Haskard M. R. 和 Curkowicz G. R.《采用三值激活函数和有限整数权重的多层神经网络的超大规模集成电路实现》。载于微电子国际会议，1995年。[15] Hinton Geoffrey、Vinyals Oriol 和 Dean Jeff。《神经网络中的知识蒸馏》，2015年。[16] Di Guglielmo Giuseppe、Duarte Javier、Harris Philip、Hoang Duc、Jindariani Sergo、Kreinar Edward、Liu Mia、Loncar Vladimir、Ngadiuba Jennifer、Pedro Kevin、Pierini Maurizio、Rankin Dylan、Sagear Sheila、Summers Sioni、Tran Nhan 和 Wu Zhenbin。《使用HLS4ML将现场可编程门阵列上的深度神经网络压缩至二值和三值精度》，2020年12月。[17] Curkowicz GR 和 Haskard MR。《一种可产生三值输出人工神经网络的反向传播改进方法，采用有限整数权重》，1994年。

[18] Alemdar Hande、Leroy Vincent、Prost-Boucle Adrien 和 P´etrot Fr´ed´eric。《面向资源高效型人工智能应用的三值神经网络》。载于*2017*年神经网络国际联合会议（*IJCNN*），第2547–2554页，2017年5月。

[19] Dbouk Hassan、Sanghvi Hetul、Mehendale Mahesh 和 Shanbhag Naresh。DBQ：面向轻量级深度神经网络的可微分支量化器，2020年7月。

[20] Kim Hyungjun、Kim Kyungsu、Kim Jinseok 和 Kim Jae- Joon。BinaryDuo：通过耦合二值激活减少二值激活网络中的梯度失配，2020年2月。

[21] Achterhold Jan、Koehler JanMathias、SchmeinkAnke 和 Genewein Tim。变分网络量化。2018年2月。

[22] Deng Jia、Dong Wei、Socher Richard、Li Li-Jia、Li Kai 和 Fei-Fei Li。ImageNet：大型分层图像数据库。载于 *2009*年 *IEEE* 计算机视觉与模式识别会议，第248–255页。IEEE，2009年。

[23] Yang Jiwei、Shen Xu、Xing Jun、Tian Xinmei、Li Houqiang、Deng Bing、Huang Jianqiang 和 Hua Xiansheng。《量化网络》，2019年11月。

[24] Faraone Julian、Fraser Nicholas、Blott Michaela 和 Leong Philip H. W. SYQ：面向高效深度神经网络的对称量化学习。第4300–4309页，2018年。

[25] Muller Lorenz K 和 Indiveri Giacomo
面向低分辨率突触权重神经网络的舍入方法，2015年。

[26] Esser Steven K，Merolla Paul A，Arthur John V，Cassidy An- drew S., Appuswamy Rathinakumar, Andreopoulos Alexander, Berg David, McKinstry Jeffrey L., Melano Timothy, Davis R., Nolfo Carmelo di, Datta Pallab, Amir Arnon, Taba Brian, Flick- ner Myron D. 和 Modha Dharmendra S.《用于快速、高能效神经形态计算的卷积网络》，2016年。

[27] 何恺明、张祥雨、任少卿和孙剑
用于图像识别的深度残差学习。载于*IEEE*计算机视觉与模式识别会议论文集，第770–778页，2016年。

[28] Simonyan Karen 和 Zisserman Andrew
用于大规模图像识别的超深卷积网络，2014年。

[29]Ando Kota、Ueyoshi Kodai、Orimo Kentaro、Yonekawa Haruyoshi、Sato Shimpei、Nakahara Hiroki、Ikebe Masayuki、Asai Tetsuya、Takamaeda-Yamazaki Shinya、Kuroda Tadahiro 和 Motomura Masato。BRein memory：一种采用 65 nm 互补金属氧化物半导体工艺的二值/三值可重构存内深度神经网络加速器，包含 13 层、4.2 K 个神经元和 0.8 M 个突触。载于*2017*年超大规模集成电路会议，第 C24–C25 页，2017年6月。

[30] Hwang Kyuyeon 和 Sung Wonyong
使用权重 +1 0 和 -1 的定点前馈深度神经网络设计。载于*2014*年 *IEEE* 信号处理系统研讨会 *(SiPS)*，第 1–6 页，2014年10月。

[31] Adrien Prost-Boucle Alban Bourge Fr´ed´eric P´etrot Hande Alemdar Nicholas Caldwell Vincent Leroy。适用于现场可编程门阵列的可扩展高性能卷积三值神经网络架构。2017年9月。

[32] Lei Deng Peng Jiao Jing Pei Zhenzhi Wu Guoqi Li
GXNOR- Net：在统一离散化框架下，使用三值权重和激活值训练深度神经网络，无需全精度存储器。2018年4月。

[33] Hou Lu 和 Kwok James T
深度网络的损失感知权重量化，2018年2月。

[34] Cavigelli Lukas 和 Benini Luca。RPR：用于训练二值和三值权重神经网络的随机分区松弛，2020年1月。[35] Marchesi M.、Benvenuto N.、Orlandi G.、Piazza F. 和 Uncini A. 使用二的幂权重设计多层神经网络。载于*1990*年 *IEEE*国际电路与系统研讨会（*ISCAS*），第 2951–2954 页，第 4 卷，1990年5月。[36] Courbariaux Matthieu、Bengio Yoshua 和 David Jean-Pierre。Binaryconnect：在传播过程中使用二值权重训练深度神经网络，2015年。[37] Parikh Neal、Boyd Stephen 等。Proximal Algorithms，2014年。[38] Shayer Oran、Levi D. 和 Fetaya Ethan。使用局部重参数化技巧学习离散权重，2018年。

[39] Kingma Durk P、Salimans Tim 和 Welling Max
变分丢弃与局部重参数化技巧，2015年。

[40] Wang Peiqi、Xie Xinfeng、Deng Lei、Li Guoqi、Wang Dong- sheng 和 Xie Yuan。HitNet：混合三值循环神经网络。载于神经信息处理系统进展，第31卷。Curran Associates Inc.，2018年。

[41] Wang Peisong 和 Cheng Jian
定点因子化网络。载于*2017*年 *IEEE*计算机视觉与模式识别会议（*CVPR*），第3966–3974页，2017年7月，檀香山，夏威夷。IEEE。

[42] Wang Peisong、Hu Qinghao、Zhang Yifan、ZhangChunjie、Liu Yang 和 Cheng Jian。面向低比特神经网络的两步量化。载于*IEEE* 计算机视觉与模式识别会议论文集，第4376–4384页，2018年。

[43] Chen Peng、Zhuang Bohan 和 Shen Chunhua
FATNN：快速且准确的三值神经网络。第5219–5228页，2021年。

[44] Gong Ruihao、Liu Xianglong、Jiang Shenghu、Li Tianxiang、Hu Peng、Lin Jiazhen、Yu Fengwei 和 Yan Junjie。可微软量化：连接全精度与低比特神经网络。第4852–4861页，2019年。

[45] Razani Ryan、Morin Gregoire、Sari Eyyub 和 NiaVahid Par-tovi。自适应二值-三值量化。第 4613–4618 页，2021年。

[46] Diwen Wan、Fumin Shen、Li Liu、Fan Zhu、Jie Qin、Ling Shao、Heng Tao Shen。TBN：具有三值输入和二值权重的卷积神经网络，2018年9月。

[47] Jain Shubham、Gupta Sumeet Kumar 和 Raghunathan Anand。TiM-DNN：用于深度神经网络的三值存内加速器，2020年7月。

[48] Boyd Stephen、Parikh Neal、Chu Eric、Peleato Borja、Eckstein Jonathan 等。通过交替方向乘子法进行分布式优化与统计学习，2011年。

[49] Gupta Suyog、Agrawal Ankur、Gopalakrishnan Kailash 和 Narayanan Pritish。有限数值精度下的深度学习。载于 国际机器学习会议，第 1737–1746 页。PMLR，2015年。

[50] Chiueh Tzi-Dar 和 Goodman Rodney M
用于三值权重神经网络的学习算法，1988年。

[51] Xu Weixiang、He Xiangyu、Zhao Tianli、Hu Qinghao、Wang Peisong 和 Cheng Jian。软阈值三值网络。第 3 卷，第 2298–2304 页，2020年7月。

[52] Sung Wonyong、Shin Sungho 和 Hwang Kyuyeon。《量化下深度神经网络的韧性》，2016年1月。

[53] Jie Ding、JunMin Wu、Huan Wu。《三均值三值量化》，2017年11月。[54] Hu Yiming、Li Jianquan、Long Xianlei、Hu Shenhua、Zhu Jia- gang、Wang Xingang 和 Gu Qingyi。《用于深度网络压缩的聚类正则化量化》。载于 *2019*年 *IEEE* 国际图像处理会议（*ICIP*），第 914–918 页，2019年9月。[55] Penghang Yin、Yingyong Qi、Shuai Zhang、Jack Xin。《使用精确近端算子训练三值神经网络》，2016年12月。[56] Penghang Yin、Zhang Shuai、Qi Yingyong 和 Xin Jack。《面向目标检测的低位宽卷积神经网络量化与训练》，2017年。[57] Bengio Yoshua、L´eonard Nicholas 和 Courville Aaron。《通过随机神经元估计或传播梯度以实现条件计算》，2013年。[58] Bai Yu、Wang Yu-Xiang 和 Liberty Edo。《ProxQuant：通过近端算子实现量化神经网络》，2018年9月。[59] Li Yue、Ding Wenrui、Liu Chunlei、Zhang Baochang 和 Guo Guodong。《TRQ：采用残差量化的三值神经网络》，2021年5月。[60] Li Yuhang、Dong Xin、Zhang Sai Qian、Bai Haoli、Chen Yuan- peng 和 Wang Wei。《RTN：重参数化三值网络》，2019年12月。[61] Nesterov Yurii。《凸问题的原始-对偶次梯度方法》，2002年。[62] He Zhezhi、Gong Boqing 和 Fan Deliang。《利用三值化权重和高准确率优化深度卷积神经网络》。载于*2019*年 *IEEE*冬季计算机视觉应用会议（*WACV*），第 913–921 页，2019年1月。[63] Lin Zhouhan、Courbariaux Matthieu、Memisevic Roland 和 Bengio Yoshua。《少乘法神经网络》。载于国际学习表征会议，2016年2月。[64] Lei Zijian 和 Lan Liang。《通过二值嵌入和三值模型系数实现内存与计算高效的核支持向量机》，2020年10月。