对于卷积神经网络的训练收敛速度的研究*

何文卿；李刚

（吉林大学 仪器科学与电气工程学院， 长春 130012；）

摘要：本研究致力于简要探索影响卷积神经网络（CNN）训练收敛速度的关键因素，特别是考虑了训练过程中使用的数据精度、Batch Size 的大小等超参数变量。通过对这些变量在不同组合下的系统研究，我们旨在揭示它们如何单独及共同影响 CNN 模型的训练效率和性能。实验结果表明，适当选择这些参数不仅可以加速训练过程，还能在不牺牲模型准确度的前提下提高模型的性能。

关键词：卷积神经网络 人工神经网络 低精度训练 收敛速率 批大小 注意力模块

Study on the Training Convergence Speed of Convolutional Neural Networks

He Wenqing； Li Gang

(College of Instrumentation and Electrical Engineering, Jilin University,Changchun 130012,China)

Abstract: This study aims to briefly explore the key factors affecting the training convergence speed of Convolutional Neural Networks (CNNs), specifically focusing on the precision of data used during training, the size of Batch Size, and the implementation of attention mechanisms. By conducting preliminary investigations into the effects of these variables under different combinations, this paper reveals how they impact the efficiency and performance of CNN models. The experimental results demonstrate that appropriate adjustments to these parameters can accelerate the training process and enhance model performance without sacrificing accuracy.

Key words: Convolutional neural network Artificial neural network Low precision training Batch size Attention module

0 前言

在深度学习领域，卷积神经网络（CNN）已成为实现图像识别、语音识别和自然语言处理等任务的关键技术。随着人工智能技术的不断进步，CNN 的研究和应用范围不断扩大，但其训练效率和收敛速度的问题仍然是当前研究的热点和难点之一。[1]本研究旨在探索不同训练条件下，特别是在高低精度、不同 Batch Size 以及是否采用注意力机制等因素对 CNN 训练收敛速度的影响，旨在为深度学习模型的快速有效训练提供理论指导和实践方案。

卷积神经网络的训练收敛速度受多种因素影响，其中训练数据的精度、Batch Size 的大小以及是否引入注意力机制等因素都对模型的训练效率和性能产生重要影响。近年来，已有研究开始关注这些因素对 CNN 训练过程的影响。现有实验表明，使用低精度数据进行训练可以在保持模型性能的同时显著提高训练速度，这为在硬件资源有限的情况下加速模型训练提供了可能。[2]此外，也有研究表明，合适的 Batch Size 可以有效地平衡模型训练的收敛速度和稳定性，对于加快模型训练过程具有重要意义。[3]但是，也有研究指出，较大的 Batch Size 可能会对最终的结果准确性和泛用性有不利影响。[4]

基于以上研究，本文还进一步探讨了引入 CBAM 注意力机制对 CNN 训练收敛速度的影响。注意力机制作为一种有效的信息筛选和重点强化手段，已被证实能够提高模型对关键信息的捕捉能力，从而提升模型的性能。[5]

本研究旨在初步探讨高低精度训练数据、不同 Batch Size 及注意力机制引入等因素对卷积神经网络训练收敛速度的影响。虽然本次探索较为基础，但提供了对不同训练条件下网络性能的初步了解，为后续深入研究和实践应用提供了一定的参考。

1 试验方法和实验方案

1.1 试验方法

使用 Batch Size 为 32 的没有加入 CBAM 注意力的 ResNet-18 网络作为标准对照组，数据精度为 FP64。低精度组使用 TF32 和 BF16；批处理实验组将 Batch Size 设置为 16 和 64；加入注意力的组为分别加入 CBAM 中通道注意力和空间注意力。

1.2 实验方案

分别使用上述网络，在 CIFAR-10 数据集上进行训练。CIFAR-10 包含 60000 张 32x32 的彩色图片，分为 10 个类。[6]将训练时产生的 Loss 绘制成为图形，依次对比分析。具体实验内容如下表所示

表 1 实验内容表

Table 1 Experiment Content Table

批处理大小实验	数据精度实验	有无注意力实验
ResNet-18,BatchSize 为 16,精度为 TF32,无注意力	ResNet-18,BatchSize 为 64,精度为 FP64,无注意力	ResNet-18,BatchSize 为 64,精度为 TF32,无注意力
ResNet-18,BatchSize 为 32,精度为 TF32,无注意力	ResNet-18,BatchSize 为 64,精度为 TF32,无注意力	ResNet-18,BatchSize 为 64,精度为 TF32,包含通道注意力和空间注意力
ResNet-18,BatchSize 为 64,精度为 TF32,无注意力	ResNet-18,BatchSize 为 64,精度为 BF16,无注意力	

2 网络模型

在网络模型部分，采用 ResNet-18 神经网络作为标准的对照组。

2.1 基础网络模型

ResNet-18 模型是深度残差网络的一种轻量级形式，设计初衷是为了解决深度网络训练中的退化问题。[7]该模型包含 18 层，核心构成为残差块，旨在通过学习输入与输出间的残差来优化训练过程，有效避免了随网络加深而导致的性能下降。[8]

模型起始于一个卷积层，用以提取图像的基础特征，随后接入批量归一化和 ReLU 激活函数进行处理。接着，模型主体由八个残差块组成，每个块内包含两层卷积，每层后均跟有批量归一化和 ReLU 激活。[9]这些卷积层的目标是捕捉输入数据的细微差异，而残差连接则保证了数据的直接流通，有助于减轻梯度消失问题，保障了即便网络深度增加，模型性能也不会降低。

在所有残差块处理完毕后，通过全局平均池化层对特征图进行空间尺寸缩减，最终通过一个全连接层将得到的特征映射为分类预测结果。

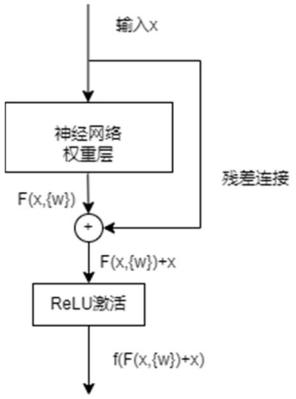

图 1 典型残差模块示意图

Fig.1 Schematic Diagram of a Typical Residual Module

残差连接详细算法步骤：

(1) 设置初始权系 $w(0)$为较小的随机非零值。

(2) 给定输入/输出样本对，计算网络的输出：

$$y_i = f(F(x,\{w_i\}) + x) \quad (1)$$

式中：y_i——在第i层残差连接网络，样本数据输入后，对应的输出数据

$F(x,\{w_i\})$——表示残差块中的权重层（例如卷积层）对输入的处理结果

f——是激励函数，采用 ReLU 型，即

$$f(z) = max(0,z) \quad (2)$$

在反向传播过程中，假设我们正在计算损失函数L关于残差块输入x的梯度$\frac{\partial L}{\partial x}$。应用链式法则，我们有

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x} \quad (3)$$

由于$y_i = f(F(x,\{w_i\}) + x)$，则有下式

$$\frac{\partial y}{\partial x} = \frac{\partial F(x,\{w\})}{\partial x} + 1 \quad (4)$$

因此，反向传播的梯度可以写为：

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \cdot \left(\frac{\partial f(F(x,\{w\}))}{\partial x} + 1\right) \quad (5)$$

注意到$\frac{\partial F(x,\{w\})}{\partial x}$可能在深度较大时非常小，但是加

上常数项1后，即使$\frac{\partial F(x,\{w\})}{\partial x}$非常小，梯度$\frac{\partial L}{\partial x}$也不会消失，因为它至少等于$\frac{\partial L}{\partial y}$。

2.2 低精度网络模型

在深度学习领域，低精度数值格式对模型的训练效率和 loss 收敛速度产生了显著影响。这些低精度格式通过减少每个数据点的位宽，可以在相同的硬件资源下加速计算过程并提高数据吞吐量。由于可以更快地执行反向传播和参数更新，模型训练周期得以缩短，从而加速了 loss 的收敛。[10]

低精度计算使得模型可以利用更高的并行度和加速矩阵运算，这对于深度学习中广泛使用的大规模矩阵乘法尤为重要。这种加速不仅提高了模型训练的效率，还有助于在更短的时间内探索更广泛的参数空间，优化模型结构和超参数配置。

尽管低精度可能会引起一定程度的数值精度下降，但多项研究和实践表明，深度学习模型通常对此具有鲁棒性。[11,12]特别是通过采用混合精度训练策略——在关键计算步骤中保持高精度以维护数值稳定性，而在其他步骤中采用低精度以提高计算效率——可以在保持模型性能的同时显著加速训练过程。

FP64（64 位双精度浮点）提供最高的数值精度，适用于精确计算要求的应用。它包含 1 位符号位、11 位指数和 52 位尾数。高精度确保数值计算的准确性，但增加了存储和计算资源的需求。

TF32（TensorFloat-32）是专为 NVIDIA 的 Ampere 架构 GPU 设计的，旨在优化深度学习训练。[14]它使用 FP32 的指数范围和与 BF16 相同数量的尾数位数（10 位），结合了 1 位符号位和 8 位指数。TF32 提供了与 FP32 相近的精度，同时显著提高了计算性能。

BF16（16 位截断浮点数）旨在优化深度学习的计算效率，通过保持与 FP32 相同的 8 位指数范围但仅使用 7 位尾数（加 1 位隐藏位），减少了数据格式的精度。BF16 的设计允许它在不牺牲太多数值范围的情况下，提高计算速度，特别是在深度学习的矩阵乘法操作中。[13]

通过 Pytorch 框架中输入数据的类型，即可直接改变模型和输入数据的精度。而且，根据对 Nvidia 的 CUDA 设备研究[15]，低精度的数据，如 TF32 和 BF16，在进行通用矩阵乘法运算(GEMM)时，可以在硬件层面使用更高效率的逻辑单元，提高运算效率。

2.3 数据批大小

在深度学习领域，批量大小（Batch Size）的选择对模型训练过程及其收敛速度产生深远影响。批量大小定义了每次迭代中用于计算梯度和更新模型参数的样本数目，其设置不仅影响训练过程中的内存需求，还直接关联到模型训练的稳定性、收敛效率以及最终模型的泛化能力。

较小的批量大小能够促进模型频繁更新，从而增强模型在每个训练周期（Epoch）内的学习机会。这种策略通过增加训练过程的随机性，有助于模型避免陷入局部最优解，潜在地提升最终模型的泛化表现。[17]在横向的故障诊断应用场景下，能够发挥出不俗的表现。[16]但是，小批量训练的代价是增加了梯度更新的方差，可能会引入训练过程的不稳定性，延长模型达到收敛状态所需的训练周期。

与之相对，较大的批量大小通过减少训练过程中的随机波动，有助于梯度更新的稳定性，从而加速模型向最优解的收敛。[4]此外，大批量训练能够更有效地利用现代硬件平台（例如 GPU 和 TPU）的高并行性，显著缩短每个训练周期的时间。然而，大批量训练也存在潜在的缺点，包括对内存资源的高需求以及可能导致模型收敛至次优解的风险。[18]

2.5 注意力机制

在现代深度学习架构中，卷积块注意力机制（CBAM）的引入代表了对模型内在表示能力优化的一种尝试，尤其在易用性方面表现出独特的优势。[5,19]CBAM 通过空间和通道维度的注意力聚焦，精细调整网络对输入数据的解析力度，使得网络能够更加高效地识别并利用对分类或回归任务至关重要的特征信息。

具体到 ResNet-18 这一轻量级网络结构，CBAM 的融合不仅增强了模型对关键信息的提取能力，也间接影响了模型的收敛速度。[20]通过对输入特征进行动态加权，CBAM 可能有助于模型在早期训练阶段就能够更加准确地捕捉到数据中的关键模式，从而在较少的迭代次数内实现有效的性能提升。

然而，CBAM 引入的优化并非没有成本。增加的

可训练参数意味着对计算资源的额外需求，尤其是在参数调优和模型训练的初期阶段。[21]CBAM 对于加速特定深度网络结构的收敛具有显著效果，特别是在处理高度复杂的数据集时，其能力在显著减少训练所需迭代次数的同时，保持或甚至提高了模型的准确率。

未来的研究可以进一步探索 CBAM 在不同卷积神经网络模型中的应用效果，以及如何平衡其带来的额外计算开销与收敛速度提升之间的关系，以实现更加高效和精准的深度学习模型训练的同时，能够在更少的迭代次数后达到更佳的效果。

3 结果与比较

3.1 低精度网络模型比较

在训练过程中，Loss 值变化整体趋势相似，但具体收敛速率上，TF32 和 FP64 速率相似，而 BF16 收敛速率始终较为缓慢，且在 3000 次迭代后几乎不进一步收敛。而对于 TF32 和 FP64 两种精度，可以观察到 FP64 的收敛过程中稳定性略优。

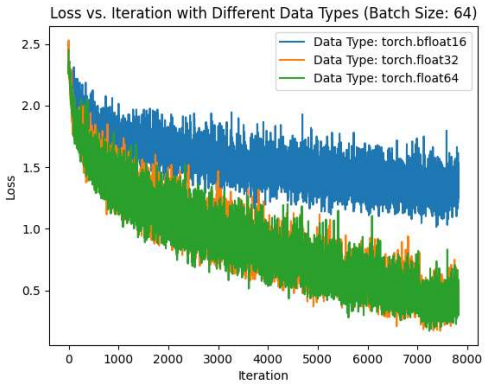

图 2 不同数据精度迭代次数-Loss 关系图

Fig.2 Loss versus Number of Iterations With Different Data Precision

注：torch.float32 为 TF32 数据精度而非 FP32

3.2 相同网络模型不同 Batch Size 比较

在训练过程中，使用 16，32 和 64 三个大小，在每一个 Epoch 数据量相同时，对应的迭代次数呈反比。

可以观察到随着迭代次数的增加，无论 Batch Size 为多少，收敛速率均逐渐下降。对于前 1000 个迭代，其收敛速率和最终的 Loss 结果大体相当。

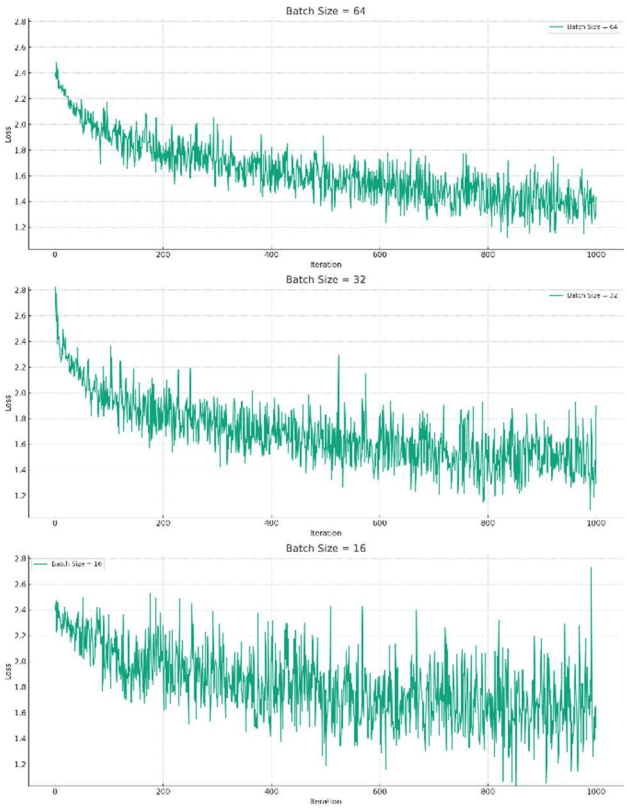

图 3 前 1000 次迭代次数-Loss 关系图

Fig.3 Graph of Loss versus Number of Former 1000th Iterations

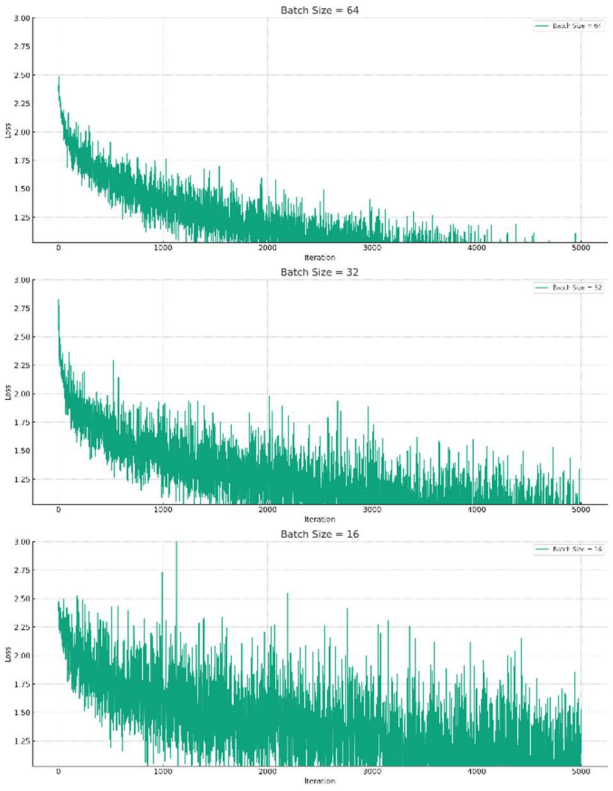

图 4 前 5000 次迭代次数-Loss 关系图

Fig.4 Graph of Loss versus Number of Former 5000th Iterations

当迭代次数进一步上升到 5000 时，相同迭代次数时，Batch Size 越小，其最终收敛得到的 Loss 值越大。

考虑到一个 Epoch 表示遍历所有数据，对应大体相同的计算量，更大的 Batch Size 需要的迭代次数更少，且 Batch Size 与所需的迭代次数呈反比。当考察相同的计算量的时候，即图中的迭代次数与 Batch Size 为反比时，最终收敛得到的 Loss 结果都在 1.0 附近。且 Batch Size 越小，其 Loss 的抖动也越大。

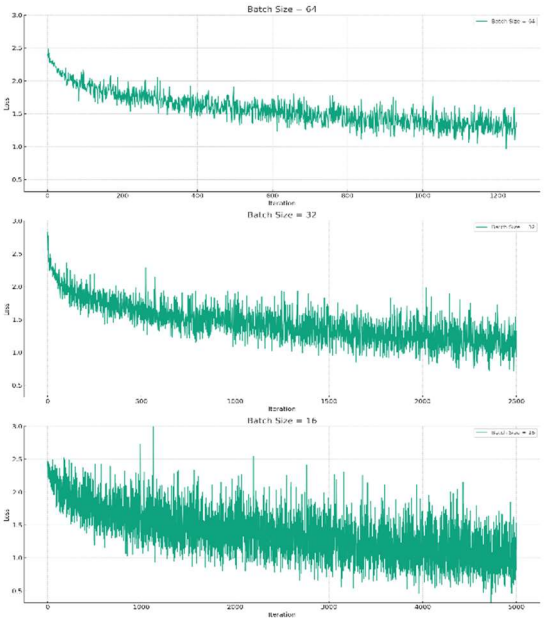

图 5　计算量-Loss 关系图

Fig.5 Graph of Loss versus Calculation

3.3 相同网络模型是否有额外注意力比较

首先，将 CBAM 注意力加入到 ResNet-18 网络的 layer1 中，从参数量上来看原本的参数量为 11,181,642 个，添加后为 11,690,634 个，相对增加量不超过 5%。

从整体上来看，只在 layer1 中加入 CBAM 注意力对于网络训练时 Loss 收敛速率的影响较小，加入 CBAM 注意力后，Loss 收敛速率的优化在 3%以内。

为了排除开始的初始化可能对网络 Loss 值的影响，早期阶段从第 200 个迭代开始计算。在 200-1500 次迭代的模型训练早期阶段中，使用 CBAM 注意力后，平均 Loss 值为 1.4221，标准差为 0.1925，而没有使用注意力的基准模型平均 Loss 值为 1.4503，标准差为 0.1852。

而在 1500-3000 次迭代的模型训练中期阶段，加入 CBAM 的模型平均 Loss 为 1.0692，标准差为 0.1493。没有加入 CBAM 的基准模型平均值为 1.0814，标准差为 0.1538。二者的差距有所缩小。而在 3000 到 5000 次迭代，二者的 Loss 值收敛速率差距在 2%以内，几乎可以认为是偶然性的细微误差。

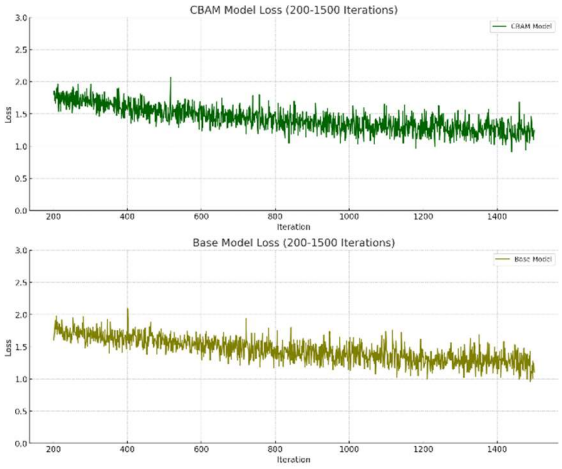

图 6　200-1500 次迭代次数-Loss 关系图

Fig.6 Graph of Loss versus Number of 200th- 1500th Iterations

针对两种网络的收敛速率在实验中没有太大差别的情况，可能是 CBAM 模块增加过少导致，故将 CBAM 模块增加到 ResNet-18 的每一个层上。此时，网络的参数量进一步增加到 11,776,944 个，相较于没有 CBAM 的网络，参数量提升了 5.3%，进行新一轮的实验。

可以观察到在 200-1500 次迭代的早期阶段，加入 CBAM 注意力的网络平均 Loss 为 1.3974，没有加入的对照组基准网络 Loss 值为 1.4461。而在 1500-3000 次，加入 CBAM 注意力的网络平均 Loss

为 1.0222，而没有注意力的网络的 Loss 值为 1.0903。改善的 Loss 大于其参数量相对于原本增加的值且二者的 Loss 标准差均为 0.1544。说明在中期阶段，加入 CBAM 能够加速网络的收敛。

在 3000-5000 次迭代时，实验组 CBAM 网络的 Loss 进一步降低为 0.7423，而对照组为 0.82363。二者绝对值差距几乎没有缩小，而在之后的 5000-8000 次迭代中，二者 Loss 的绝对值差距也没有减小。说明注意力机制在迭代次数较多时，也能发挥出效果。

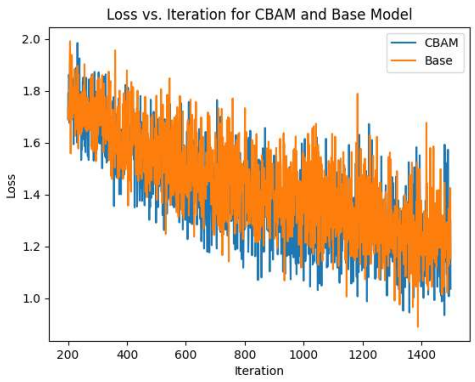

图 7　200-1500 次迭代次数-Loss 关系图

Fig.7 Graph of Loss versus Number of 200th- 1500th Iterations

4　结论与展望

通过对训练数据精度，批处理大小和有无额外注意力模块在 ResNet-18 网络上的研究，找到了在次情况下的一些简单规律。在精度上，使用 TF32 半数据精度可以在 Batch Size 相同时，训练到相同 Epoch 时，取得与 FP64 全精度数据非常类似的效果，而如果使用了 BF16 则可能很难在相同的 Epoch 下得到相同的效果，甚至可能最终难以收敛。在批处理大小 Batch Size 上，相同的计算量在不同的大小上最终结果相当，而较大的批处理大小能够减少 Loss 的抖动，当不考虑计算量时，较大的批处理大小能够在相同迭代次数取得更小的 Loss 值。

本文也有一些局限性，如只使用了 ResNet-18 进行了训练，没有考虑更大规模的网络；只使用了 CIFAR-10 这一个数据集，没有考虑到更大规模的数据或者更加复杂的分类情况；没有考虑更大的 Batch Size；没有能够解释不同 Batch Size 训练相同计算量时为什么较大的 Batch Size 能够抑制它的 Loss 值抖动；没有解释为什么更多层次的 CBAM 注意力能够相较于更少层次的注意力的改进，是因为参数增加还是在某一些关键部位增加导致的。

目前，已经存在对于可变批处理大小的研究[18]，训练框架也能够在单一网络中使用不同的数据精度[23,24]，进行混合精度训练。基于现有研究和本研究，可以进一步探索在维持现有 Loss 最终结果基本不变的前提下，如何进一步提高收敛速率，亦可以探索在维持现有 Loss 收敛速率大体不变的前提下，如何通过更改 Batch Size 或数据精度以减少计算量，减少内存开销。

参考文献

[1] Ye X, Dai P, Luo J, et al. Accelerating CNN training by pruning activation gradients[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16. Springer International Publishing, 2020: 322-338.

[2] Gupta S, Agrawal A, Gopalakrishnan K, et al. Deep learning with limited numerical precision[C]//International conference on machine learning. PMLR, 2015: 1737-1746.

[3] Mirzadeh S I, Farajtabar M, Pascanu R, et al. Understanding the role of training regimes in continual learning[J]. Advances in Neural Information Processing Systems, 2020, 33: 7308-7320.

[4] Kandel I, Castelli M. The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset[J]. ICT express, 2020, 6(4): 312-315.

[5] Chen B, Zhang Z, Liu N, et al. Spatiotemporal convolutional neural network with convolutional block attention module for micro-expression recognition[J]. Information, 2020, 11(8): 380.

[6] Abouelnaga Y, Ali O S, Rady H, et al. Cifar-10: Knn-based ensemble of classifiers[C]//2016 International Conference

on Computational Science and Computational Intelligence (CSCI). IEEE, 2016: 1192-1195.

[7] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[8] He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. Springer International Publishing, 2016: 630-645.

[9] Bjorck N, Gomes C P, Selman B, et al. Understanding batch normalization[J]. Advances in neural information processing systems, 2018, 31.

[10] Micikevicius P, Narang S, Alben J, et al. Mixed precision training[J]. arXiv preprint arXiv:1710.03740, 2017.

[11] De Sa C, Leszczynski M, Zhang J, et al. High-accuracy low-precision training[J]. arXiv preprint arXiv:1803.03383, 2018.

[12] Fujita K, Yamaguchi T, Kikuchi Y, et al. Calculation of cross-correlation function accelerated by TensorFloat-32 Tensor Core operations on NVIDIA's Ampere and Hopper GPUs[J]. Journal of Computational Science, 2023, 68: 101986.

[13] Osorio J, Armejach A, Petit E, et al. A BF16 FMA is all you need for DNN training[J]. IEEE Transactions on Emerging Topics in Computing, 2022, 10(3): 1302-1314.

[14] Choquette J, Lee E, Krashinsky R, et al. 3.2 the a100 datacenter gpu and ampere architecture[C]//2021 IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2021, 64: 48-50.

[15] Abdelkhalik H, Arafa Y, Santhi N, et al. Demystifying the nvidia ampere architecture through microbenchmarking and instruction-level analysis[C]//2022 IEEE High Performance Extreme Computing Conference (HPEC). IEEE, 2022: 1-8.

[16] Yao Y, Wang J, Long P, et al. Small‐batch‐size convolutional neural network based fault diagnosis system for nuclear energy production safety with big‐data environment[J]. International Journal of Energy Research, 2020, 44(7): 5841-5855.

[17] Masters D, Luschi C. Revisiting small batch training for deep neural networks[J]. arXiv preprint arXiv:1804.07612, 2018.

[18] Devarakonda A, Naumov M, Garland M. Adabatch: Adaptive batch sizes for training deep neural networks[J]. arXiv preprint arXiv:1712.02029, 2017.

[19] Xiao J, Li Z, Yang L, et al. Patch-wise mixed-precision quantization of vision transformer[C]//2023 International Joint Conference on Neural Networks (IJCNN). IEEE, 2023: 1-7.

[20] Chen L, Yao H, Fu J, et al. The classification and localization of crack using lightweight convolutional neural network with CBAM[J]. Engineering Structures, 2023, 275: 115291.

[21] Luo Y, Wang Z. An improved resnet algorithm based on cbam[C]//2021 International Conference on Computer Network, Electronic and Automation (ICCNEA). IEEE, 2021: 121-125.

[22] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[23] Cai Z, Vasconcelos N. Rethinking differentiable search for mixed-precision neural networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 2349-2358.

[24] Hayford J, Goldman-Wetzler J, Wang E, et al. Speeding up and reducing memory usage for scientific machine learning via mixed precision[J]. arXiv preprint arXiv:2401.16645, 2024.