Advances in Applied Mathematics 应用数学进展, 2024, 13(6), 2675-2686
Published Online June 2024 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2024.136256

Hans汉斯

基于梯度下降的循环批量权重平均优化算法

赵子锐，李晨龙*，杨卫华

太原理工大学数学学院，山西 太原

收稿日期：2024年5月21日；录用日期：2024年6月15日；发布日期：2024年6月24日

摘　要

神经网络模型的权重对模型的性能具有关键影响，权重的更新方式主要通过梯度下降算法实现。随机权重平均算法改进了权重更新方法，它通过平均随机梯度下降过程中得到的多个权重样本提高模型的泛化能力。然后，该算法并没有对平均后的模型进一步训练，也没有参与并影响模型的训练过程。本文针对该算法的局限性，结合已有的循环随机权重平均算法，提出循环批量权重平均算法。与循环随机权重平均算法在训练周期之间平均不同，本文所提算法在每个训练周期内的各个批次之间进行权重平均，并将平均后的权重作为下一个批次训练的初始权重，循环融合权重的平均过程和更新过程。本文将提出的算法与梯度下降和随机权重平均算法在CIFAR-10和CIFAR-100数据集上分别对VGG-16和ResNet-18模型进行多次仿真实验，并对实验结果进行比较。实验结果表明，循环批量权重平均算法显著加快模型的收敛速度，提升模型的训练效率，并能提高模型在测试集上的准确率。

关键词

深度神经网络，梯度下降，循环批量权重平均

Recurrent Batch Weight Averaging Optimization Algorithm Based on Gradient Descent

Zirui Zhao, Chenlong Li*, Weihua Yang

School of Mathematics, Taiyuan University of Technology, Taiyuan Shanxi

Received: May 21st, 2024; accepted: Jun. 15th, 2024; published: Jun. 24th, 2024

Abstract

The weights of the neural network model have a key impact on the performance of the model, and

*通讯作者。

文章引用: 赵子锐, 李晨龙, 杨卫华. 基于梯度下降的循环批量权重平均优化算法[J]. 应用数学进展, 2024, 13(6): 2675-2686. DOI: 10.12677/aam.2024.136256

the update method of the weights is mainly realized by the gradient descent algorithm. The stochastic weight averaging algorithm improves the weight updating method, which improves the generalization ability of the model by averaging multiple weight samples obtained in the process of stochastic gradient descent. Then, the algorithm does not further train the averaged model, nor does it affect the training process of the model. Aiming at the limitations of this algorithm, this paper proposes a recurrent batch weight averaging algorithm based on the existing recurrent random weight averaging algorithm. Different from the recurrent random weight averaging algorithm which averages between training epoch, the proposed algorithm averages the weights between batch in each training epoch, uses the averaged weights as the initial weights of the next batch of training, and circulates the average process and updates process of the weights. In this paper, the proposed algorithm is compared with gradient descent and stochastic weight averaging algorithm on the CIFAR-10 and CIFAR-100 datasets for multiple simulation experiments on the VGG-16 and ResNet-18 models, respectively, and the experimental results are compared. The experimental results show that the cyclic batch weight averaging algorithm can significantly accelerate the convergence speed of the model, improve the training efficiency of the model, and improve the accuracy of the model on the test set.

Keywords

Deep Neural Networks, Gradient Descent, Recurrent Batch Weight Averaging

1. 引言

在深度神经网络训练过程中，梯度是用来指导模型权重更新的关键信息，通过对损失函数的梯度进行反向传播来计算[1]。在理想情况下，梯度应该准确地反映模型输出与目标值之间的差异。随机梯度下降算法[2] (Stochastic Gradient Descent, SGD)及其各种优化算法是训练神经网络的实用方法。但是由于SGD 方差较大，收敛速度较慢，为了改善这一问题，许多方法通常在一定时间后计算全梯度以减少方差。例如，随机平均梯度(stochastic average gradient, SAG)算法在每次迭代中计算单个样本的梯度并将其储存起来，然后使用所有存储的梯度的平均值来更新参数，加快收敛速度[3]。随机方差缩减梯度[4] (stochasticvariance reduction gradient, SVRG)算法通过定期计算全数据集上的精确梯度来修正某一个样本或一小批样本的随机梯度，以减少因梯度估计不准确而引入的额外方差。目前已经涌现出许多改进的方差缩减方法，如随机双坐标上升[5] (Stochastic dual coordinate ascent, SDCA)、随机递归梯度[6] (Stochastic Recursive Gradient, SARAH)、半随机梯度下降[7] (Semi-Stochastic Gradient Descent, S2GD)等。文献[8]总结了随机梯度下降－上升(Stochastic Gradient Descent-Ascent, SGDA)方法，并提出 SGDA 的几种新变体，例如新的方差减少方法(L-SVRGDA)以及一种具有坐标随机化的新方法(SEGA-SGDA)。然而在实际应用中，梯度噪声是一个普遍存在的问题，它指的是由于数据的随机抽样、数据本身的噪声和模型的复杂性等而引入的梯度计算误差。在使用小批量梯度下降法时，每个批次随机选择的样本会导致梯度估计噪声[9]。适当的梯度噪声可以帮助神经网络避免陷入局部最小值或鞍点，促进全局搜索能力，从而可能找到更好的最优解。如果噪声太大或控制不恰当，可能会影响训练过程，使得模型难以收敛，导致性能下降。文献[10]通过研究 SGD 的梯度噪声，量化了有效梯度噪声的大小，并指出噪声对算法的影响。文献[11]

指出随机梯度中的噪声在一定程度上可以增强模型的泛化能力。文献[12]分析了随机梯度噪声的一些常见属性，以及它们如何影响基于 SGD 的优化。

随机权重平均[13] (Stochastic Weight Averaging, SWA)通过采用修改后的周期性或高常数学习率策略，对梯度下降遍历的模型权重进行尾部平均[14]，生成新模型的权重，从而有效提高模型的泛化能力。SWA 改进了权重的更新方法，但是并没有对更新后的模型继续训练，也没有直接作用于训练过程中的梯度噪声，因此对加速模型训练过程和提升收敛速度方面几乎没有影响。在之前的研究中，我们提出了一种循环随机权重平均算法(Recurrent Stochastic Weight Averaging, RSWA)，循环随机权重平均只是在各个训练周期之间平均模型的权重并继续训练，忽视了同一个训练周期内的各个批次之间的信息差异。因此，针对 SWA 算法的局限性和 RSWA 的不足，本文提出循环批量权重平均算法(Recurrent Batch Weight Averaging, RBWA)，该算法通过在每个训练周期(epoch)中的各个批次(batch)之间平均使用梯度下降更新的权重，并将平均后的权重作为下一个 batch 的模型初始权重继续训练，如此循环，直至训练结束。RBWA 旨在直接作用于训练过程中的有害梯度噪声，通过循环平均的方式减少它们对权重的影响，优化权重更新的方向。

本文在 CIFAR-10 和 CIFAR-100 数据集上分别对 VGG-16 和 ResNet-18 模型进行仿真实验，在实验中通过改变批量大小和平均公式对算法性能进行探讨。实验结果表明，循环批量权重平均算法显著加快模型的训练速度和收敛速度，损失值更低且方差更小，在测试集上的准确率具有明显提升，有利于提高模型的泛化能力和鲁棒性。

2. 预备知识

随机权重平均算法

近年来，研究者提出了一种随机权重平均(Stochastic Weight Averaging, SWA)方法[13]，并受到了广泛关注。随机权重平均有两层含义，一方面表示它是随机梯度下降生成的权重的平均值；另一方面是指在周期性或恒定的学习率下，随机梯度下降被看成近似地从深度神经网络的损失面采样，从而产生随机权重。与 SGD 相比，SWA 可以显著提高模型的泛化能力，并且没有增加额外的计算开销，也为集成到现有的优化框架提供了便利性。

Algorithm 1 随机权重平均 (SWA)

Input: 初始权重 $\hat{W}$，由 α_1, α_2 控制的学习率策略 LRS，周期长度 c（对于常数学习率 $c = 1$），迭代次数 n

Output: 平均权重 W_{SWA}

$W \leftarrow \hat{W}$ // 初始化权重
$W_{SWA} \leftarrow W$
for $i = 0, 1, \ldots, n$ **do**
  $\alpha \leftarrow \alpha(i)$ // 根据 LRS 策略计算学习率
  $W \leftarrow W - \alpha \nabla f_i(W)$ // 随机梯度下降
  if $\mathrm{mod}(i, c) = 0$ **then**
    $n_{\text{models}} \leftarrow i/c$
    $W_{SWA} \leftarrow (n_{\text{models}} \cdot W_{SWA} + W)/(n_{\text{models}} + 1)$ // 平均更新公式
  end if
end for

Figure 1. Stochastic weight averaging algorithm [13]
图 1. 随机权重平均算法[13]

在训练的初始阶段，使用常规的 SGD 或其他变体优化算法，直到 SGD 接近收敛，然后开始执行 SWA 策略。此时学习率设置为一个周期性变化的函数或者是一个较高的常数值，重新沿着 SGD 的轨迹对损失函数 $f(w)$ 的局部最优解 W_{SGD} 进行采样，最后执行权重平均操作，该操作输出最优解的平均值 W_{SWA} 作为最终输出，其中 W 表示神经网络模型的权重。图 1 展示了实现 SWA 的伪代码。

3. 循环批量权重平均算法

由于 SWA 算法的平均过程并没有直接参与模型的训练过程，且 RSWA 仅仅只是在训练过程的各个 epoch 之间进行权重平均和更新，因此，本文提出了循环批量权重平均算法(RBWA)，该算法旨在通过对每个训练周期内各个 batch 之间的权重更新进行平均，并使用平均后的权重继续训练，直接降低梯度噪声的影响。该算法不仅考虑了单个批次内的数据特性，还整合了整个周期内所有批次的累积信息，从而在权重更新过程中引入更多的稳定性和准确性。通过这种方式，RBWA 算法能够有效平滑梯度估计，减少因数据随机性和模型复杂性导致的有害噪声，进而提高模型的训练效率和泛化能力。本节将分别介绍使用算术平均时的 RBWA 和使用指数加权移动平均公式的 RBWA (RBWA-E)，并探讨了不同的批量大小对算法性能的影响。

3.1. 使用算术平均的循环批量权重平均算法

RBWA 从训练初期开始，使用与 SGD 相同的学习率策略。在每个 batch 结束时，对使用 SGD 更新的模型权重 W 进行采样，然后使用算术平均的更新公式执行权重平均操作，该操作输出 W_{RBWA}，并将平均后的权重作为下一个 batch 的初始权重，继续使用 SGD 训练。最终输出由 W_{RBWA} 训练后得到的模型权重 W。如此循环往复，直至训练结束，算法的整体流程如图 2 所示。

Algorithm 2 循环批量权重平均 (RBWA)

Input: 随机梯度下降权重 W_{SGD}，学习率 α，开始平均的周期次数 c，批量大小 b，数据样本大小 m，迭代周期次数 n

Output: 模型权重 W

$W \leftarrow W_{SGD}$
$W_{RBWA} \leftarrow W$
for $i = 0, 1, \ldots, n$ **do**
  $W \leftarrow W - \alpha \nabla f_i(W)$ // 随机梯度下降
  if n $\geq c$ **then**
    for $j = 0, 1, \ldots, m/b$ **do**
      $n_{\text{models}} = (n - c) \cdot j$
      $W_{RBWA} = (n_{\text{models}} \cdot W_{RBWA} + W)/(n_{\text{models}} + 1)$ // 平均更新公式
      $W \leftarrow W_{RBWA}$
    end for
  end if
end for
return W

Figure 2. Recurrent batch weight averaging algorithm
图 2. 循环批量权重平均算法

在算法 2 中，平均更新公式使用算术平均

$$W_{RBWA} = \frac{n_{models} \cdot W_{RBWA} + W}{n_{models} + 1} \tag{1}$$

将平均公式展开：

$$\begin{cases} \bar{W}_1^{RBWA} = W_1 \\ \bar{W}_2^{RBWA} = \dfrac{(W_1 + W_2)}{2} \\ \bar{W}_3^{RBWA} = \dfrac{\left(2 \cdot \bar{W}_2^{RBWA} + W_3\right)}{3} = \dfrac{(W_1 + W_2 + W_3)}{3} \\ \cdots \\ \bar{W}_n^{RBWA} = \dfrac{\left[(n-1) \cdot \bar{W}_n^{RBWA} + W_n\right]}{\mathrm{n}} = \dfrac{1}{n}\sum_{i=1}^{n} W_i \end{cases} \tag{2}$$

其中，W_n 是由平均后的权重 $\bar{W}_{n-1}^{RBWA}$ 通过第 j 个 batch 训练后得到的当前模型权重。因此，每个训练批次后的权重都会被加入到平均过程中，平均后的权重都会被更新到当前的模型中继续使用 SGD 进行训练，如此循环融合权重的更新过程和平均过程，直到训练结束。

实验选用 CIFAR-10 和 CIFAR-100 两个广泛使用的计算机视觉数据集，并进行归一化处理。在划分训练集和测试集时使用不同的数据加载器分别加载训练数据和测试数据，使他们为互不可见的数据集，这样在测试泛化性能时，具有跨数据集的效果，更好地检测模型的泛化能力。在 CIFAR10 数据集上分别比较 RBWA 与 SGD 在 ResNet-18 和 VGG-16 模型上的测试准确率和损失函数下降曲线，设置数据批量大小为 64，训练周期为 50 个 epoch，RBWA 从第 10 个 epoch 开始执行，随机选取其中一组实验结果如下所示。

由图 3 可知，在 ResNet-18 模型上，RBWA 的测试准确率为 83.36%，SGD 的测试准确率为 81.98%，测试准确率提升了 1.38%；在 VGG-16 模型上，RBWA 的测试准确率为 88.82%，SGD 的测试准确率为 86.31%，测试准确率提升了 1.89%。进一步可以看出，在引入 RBWA 之后，测试准确率迅速增高，并保持稳定。与 SGD 相比，RBWA 的测试准确率在后续迭代中方差更低。

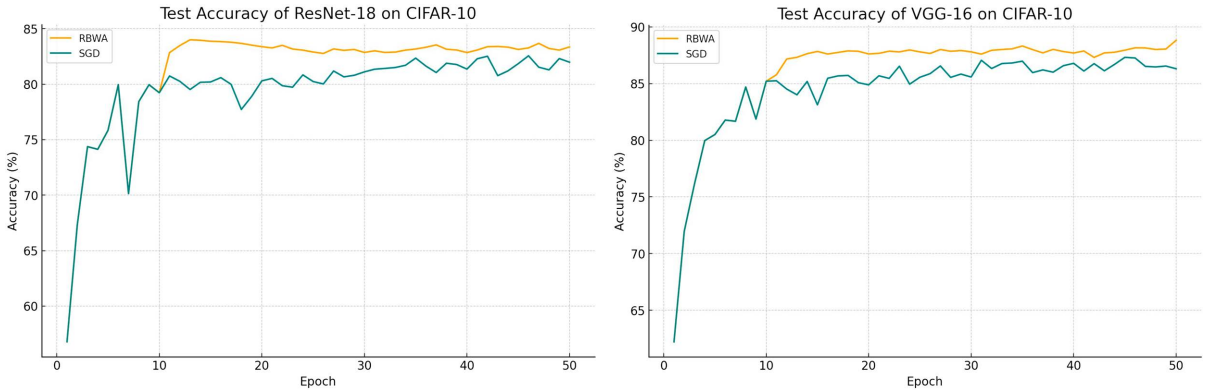

Figure 3. Test accuracy of RBWA and SGD on CIFAR-10
图 3. RBWA 和 SGD 在 CIFAR-10 上的测试准确率

由图 4 可知，RBWA 的损失收敛速度更快且能够达到更低的损失值。在损失函数接近收敛时，RBWA 的损失曲线更平稳，方差更低。具体而言，在 ResNet-18 模型上，RBWA 的最终损失值约为 0.002，SGD 的最终损失值约为 0.013；在 VGG-16 模型上，RBWA 的最终损失值约为 0.0001，SGD 的最终损失值约为 0.0076。

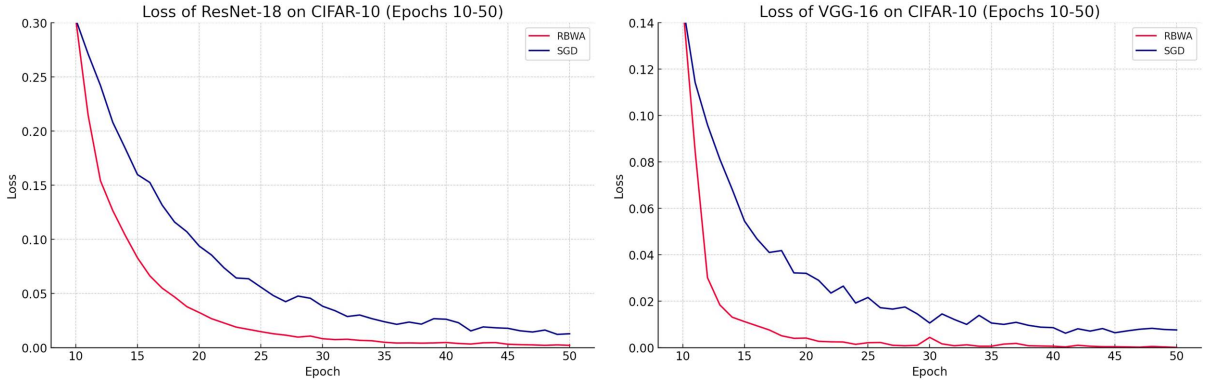

Figure 4. Loss function decline curve of RBWA and SGD on CIFAR-10
图 4. RBWA 和 SGD 在 CIFAR-10 上的损失函数下降曲线

因此，RBWA 在训练初期就能够起到显著影响，相比于 SGD，在 ResNet-18 和 VGG-16 模型上都表现出来更优秀的性能。

为了检验 RBWA 的有效性和适用性，设置数据批量大小为 64，在具有更多输出类别的 CIFAR100 数据集上分别比较 RSWA 与 SGD 在 ResNet-18 和 VGG-16 模型上的性能，随机选取其中的一组实验结果如下所示。

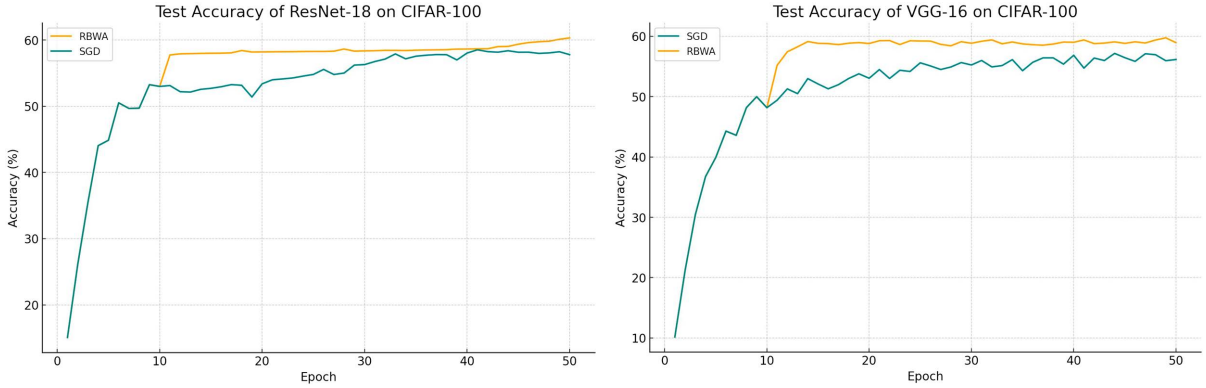

Figure 5. Test accuracy of RBWA and SGD on CIFAR-100
图 5. RBWA 和 SGD 在 CIFAR-100 上的测试准确率

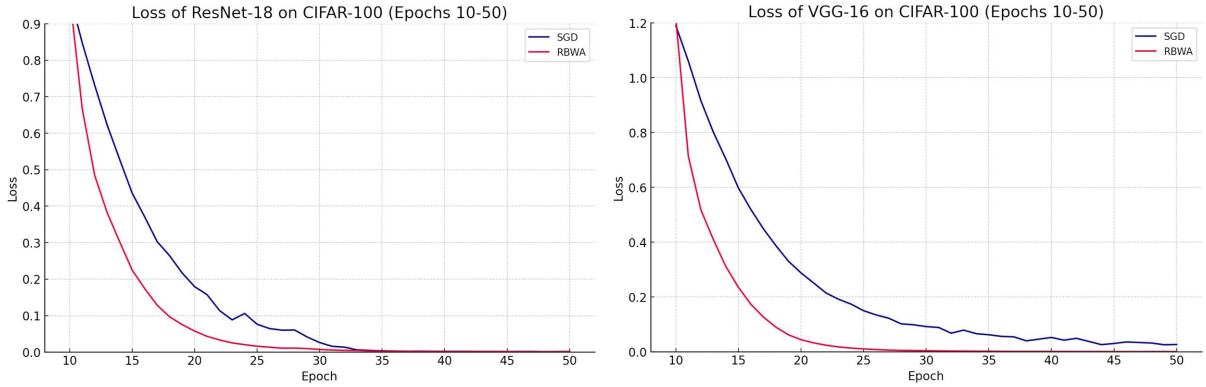

Figure 6. Loss function decline curve of RBWA and SGD on CIFAR-100
图 6. RBWA 和 SGD 在 CIFAR-100 上的损失函数下降曲线

图 5 和图 6 的结果表明，RBWA 在难度更大的 CIFAR-100 数据集上也取得了显著效果。具体而言，在 ResNet-18 模型上，RBWA 的测试准确率为 60.33%，损失值约为 0.0009，SGD 的测试准确率为 57.8%，损失值约为 0.0016；在 VGG-16 模型上，RBWA 的测试准确率为 58.95%，损失值约为 0.002，SGD 的测试准确率为 56.16%，损失值约为 0.026。RBWA 的测试准确率曲线表现出更高更稳定的准确率，损失下降曲线表现出更快的收敛速度。因此，RBWA 有效提高了模型的训练效率和泛化能力。

3.2. 不同批量大小对 RBWA 的影响

在模型训练过程中，训练集被分成多个较小的批次，每个批次包含一定数量的样本。这些批次用于在每次迭代中计算梯度和更新模型权重。批量大小的选择对权重更新及模型性能有着重要影响。在计算效率方面，较小的批量在 GPU 上可以提高训练的并行度和资源利用率；较大的批量可以提供更准确的梯度估计，但每次迭代的计算时间可能会更长。在收敛速度和稳定性方面，较小的批量可能导致梯度估计的方差增大，使得收敛速度变慢；较大的批量提供了更准确的梯度方向，但可能会陷入局部最小值。批量大小的选择是一个经验性的过程，需要考虑计算效率、模型收敛性、泛化能力和硬件限制等因素的影响。在 CIFAR-10 数据集上，保持其他实验设置不变，仅改变批量大小，对 ResNet-18 进行了多组对比实验，探究不同批量大小对 RBWA 性能的影响，实验结果如图 7 所示。

如图 7 所示，当批量大小为 16 时，RBWA 在 ResNet-18 模型上的测试准确率为 83.76%，损失值约为 0.00473；SGD 在 ResNet-18 模型上的测试准确率为 82.27%，损失值约为 0.01776。因此，相较于 SGD，RBWA 的测试准确率提升了 1.49%，损失值降低了 73.4%，损失函数的收敛速度明显加快。

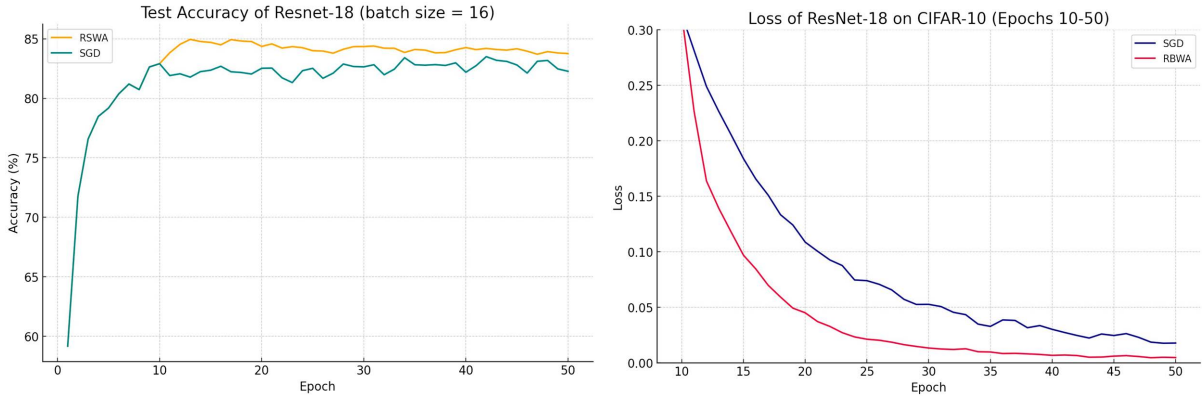

Figure 7. Accuracy and loss function decline curve for ResNet-18 with batch size 16
图 7. 批量大小为 16，ResNet-18 的准确率与损失函数下降曲线

如图 8 所示，当批量大小为 32 时，RBWA 在 ResNet-18 模型上的测试准确率为 84.25%，损失值约为 0.00384；SGD 在 ResNet-18 模型上的测试准确率为 82.5%，损失值约为 0.0176。因此，相较于 SGD，RBWA 的测试准确率提升了 1.75%，损失值降低了 78.2%，损失函数的收敛速度明显加快。

如图 9 所示，当批量大小为 128 时，RBWA 在 ResNet-18 模型上的测试准确率为 82.55%，损失值约为 0.00039；SGD 在 ResNet-18 模型上的测试准确率为 82.46%，损失值约为 0.00035。因此，相较于 SGD，RBWA 的测试准确率提升了 0.09%，提升效果微弱；损失值虽然略高于 SGD，但损失函数的收敛速度明显加快。

如图 10 所示，当批量大小为 256 时，RBWA 在 ResNet-18 模型上的测试准确率为 80.74%，损失值约为 0.00047；SGD 在 ResNet-18 模型上的测试准确率为 80.56%，损失值约为 0.00038。因此，相较于 SGD，

RBWA 的测试准确率提升了 0.18%，损失值略高于 SGD，损失函数收敛速度更快。从图 7~10 可以进一步看出，随着批次样本量的增加，SGD 在最后几个周期的结果趋于稳定，达到与 RBWA 几乎相同的效果。这可能是由于 SGD 在单次更新时所利用的样本信息更为丰富所导致的，与我们提出 RBWA 的设想相一致，即利用更丰富的样本信息提升模型效果。

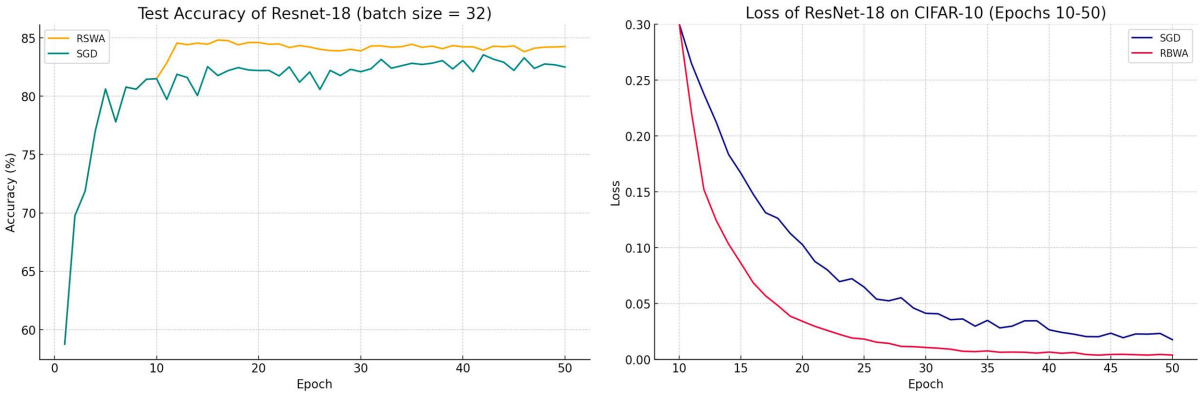

Figure 8. Accuracy and loss function decline curve for ResNet-18 with batch size 32
图 8. 批量大小为 32，ResNet-18 的准确率与损失函数下降曲线

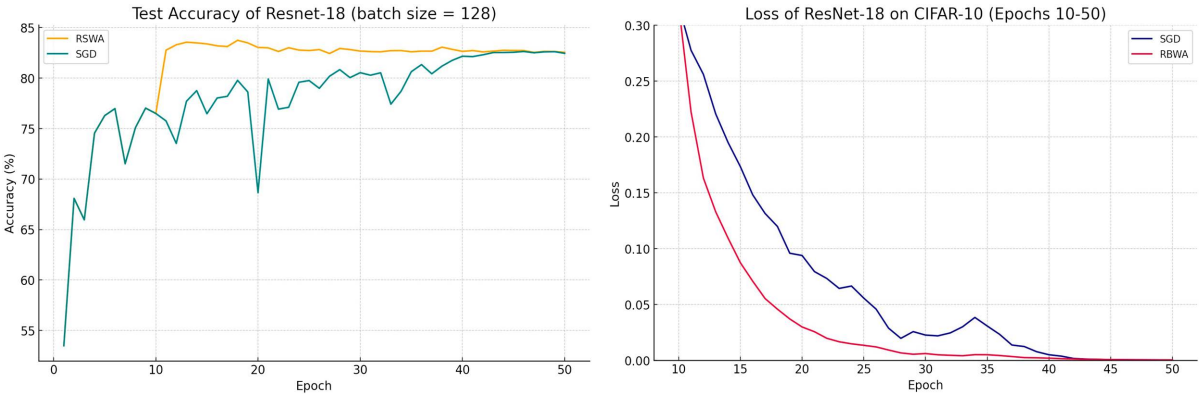

Figure 9. Accuracy and loss function decline curve for ResNet-18 with batch size 128
图 9. 批量大小为 128，ResNet-18 的准确率与损失函数下降曲线

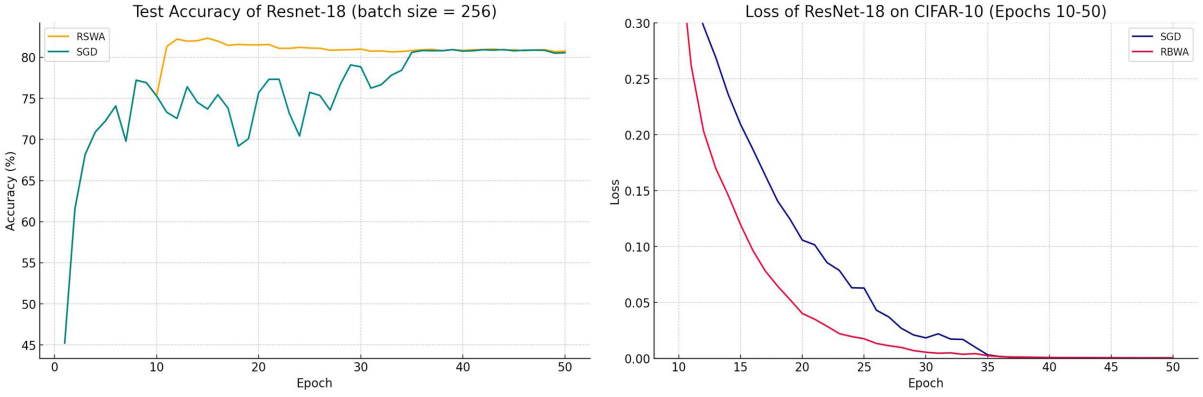

Figure 10. Accuracy and loss function decline curve for ResNet-18 with batch size 256
图 10. 批量大小为 256，ResNet-18 的准确率与损失函数下降曲线

将图 7~10 输出模型的准确率和损失值整理成表格形式，如表 1 所示。

Table 1. Effects of different batch sizes on RBWA test accuracy and loss
表 1. 不同批量大小对 RBWA 测试准确率和损失的影响

Batch size	SGD		RBWA	
	Accuracy	Loss	Accuracy	Loss
16	82.27%	0.01776	83.76%	0.00473
32	82.5%	0.01760	84.25%	0.00384
64	82.31%	0.01283	83.36%	0.00209
128	82.46%	0.00035	82.55%	0.00039
256	80.56%	0.00038	80.74%	0.00047

综上所述，对于不同的批量大小，相较于 SGD，RBWA 都能加速损失函数收敛，提高训练效率，并且能提高训练过程中的模型测试准确率。然而，RBWA 训练结束时的最终性能受到 SGD 最终性能的影响，特别是当批量大小为 128 或 256 时，RBWA 训练结束时的最终性能受到的影响最为严重。在 CIFAR-10 数据集上，对 ResNet-18 模型使用 RBWA 方法的最优批量大小为 32。

3.3. 使用指数加权移动平均的循环批量权重平均

在图 2 中，算法 2 的平均更新公式只使用了简单的算术平均。为了增强时效性，减少早期权重对模型的影响，在平均过程中可以使用指数加权移动平均(Exponential Weighted Moving Average, EWMA) [15] 更新权重。在神经网络的训练过程中，EWMA 常用于平滑损失函数的曲线或者调整学习率(如 Adam 优化算法)。与简单的算术平均不同，EWMA 在计算当前平均值时为最近的观测值分配更高的权重，而对比较旧的数据则分配递减的权重。这种指数衰减的加权机制使得 EWMA 能够更加灵敏地反映数据的最近变化，同时还保持一定程度的平滑效果，减少随机波动的影响。EWMA 的计算公式为：

$$V_t = \beta V_{t-1} + (1-\beta)\theta_t \tag{3}$$

其中，$t \geq 1$，$0 < \beta < 1$，V_t 是在时间点 t 的指数加权移动平均值，θ_t 是在时间点 t 的实际观测值，β 是衰减因子，用于控制历史观测值的权重衰减速度。衰减因子越接近 1，历史数据的影响就越大，平滑效果就越强；反之，衰减因子越小，最近的数据影响就越大，从而更能反应最新的变化趋势。

在算法 2 中，平均更新公式使用指数加权移动平均公式

$$W_{RBWA} = \beta \cdot W_{RBWA} + (1-\beta) \cdot W \tag{4}$$

当 $\beta = 0.9$ 时，将公式(4)展开：

$$\begin{cases} \hat{W}_1^{RBWA} = W_1 \\ \hat{W}_2^{RBWA} = 0.9 \cdot W_1 + 0.1 \cdot W_2 \\ \hat{W}_3^{RBWA} = 0.9 \cdot \hat{W}_2^{RBWA} + 0.1 \cdot W_3 = 0.1 \cdot W_3 + 0.1 \cdot 0.9 \cdot W_2 + 0.9^2 \cdot W_1 \\ \cdots \\ \hat{W}_n^{RBWA} = 0.9 \cdot \hat{W}_{n-1}^{RBWA} + 0.1 \cdot W_n = 0.1 \cdot W_n + 0.1 \cdot 0.9 \cdot W_{n-1} + \cdots + 0.9^n \cdot W_1 \end{cases} \tag{5}$$

其中，W_n 是由平均后的权重 $\bar{W}_{n-1}^{RBWA}$ 通过第 j 个 batch 训练后得到的当前模型权重。在指数加权移动平均的的处理过程中，通过对历史权重以指数级别的衰减，使得模型在训练过程中能够更加平滑地调整其权重，在一定程度上避免了由于数据中的高频噪声或异常值导致的剧烈波动。它不仅有效减小计算过程中的梯度噪声，还增强了模型对新数据的敏感度。通过指数移动加权平均，RBWA 能够在累积的历史权重和当前的模型权重之间找到合适的平衡，从而优化整体的学习路径。

在CIFAR-10数据集上对SGD和RSWA-E在ResNet-18和VGG-16模型上的性能表现进行对比实验，批量大小为32，训练周期为50个epoch，RBWA-E从第10个epoch开始执行，保持其它实验设置完全一致。在实验结果中随机抽选一组，结果如下所示。

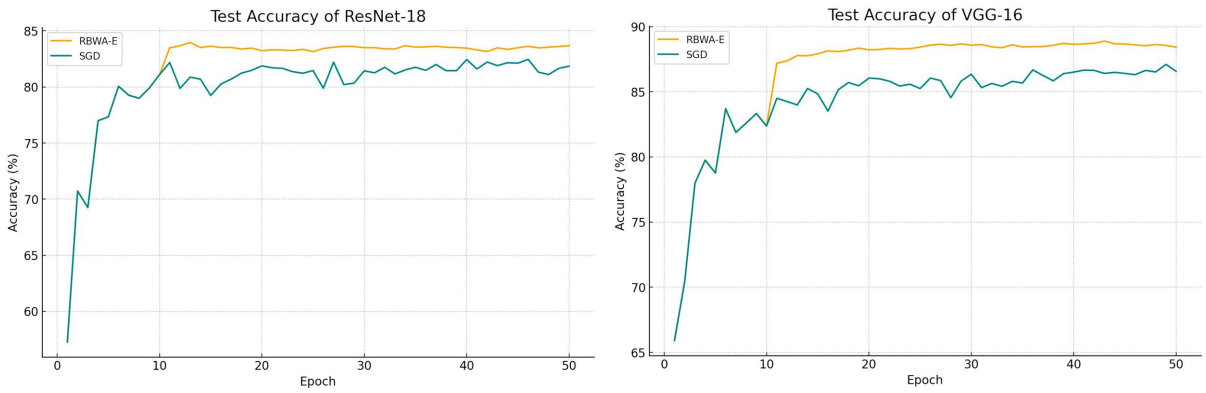

Figure 11. Test accuracy of RBWA-E and SGD on CIFAR-10
图 11. RBWA-E 和 SGD 在 CIFAR-10 上的测试准确率

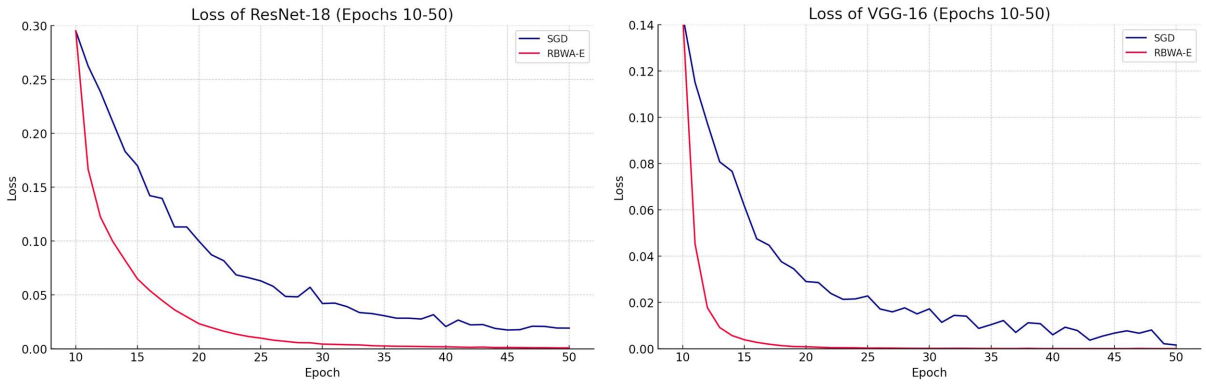

Figure 12. Loss function decline curve of RBWA-E and SGD on CIFAR-10
图 12. RBWA-E 和 SGD 在 CIFAR-10 上的损失函数下降曲线

由图11和图12可知，RBWA-E在ResNet-18和VGG-16模型上的准确率分别达到了83.68%和88.42%，相比于SGD提升了2%左右。RBWA-E还有利于加快损失函数收敛速度，并最终收敛到损失更低、方差更小的最优解，在一定程度上提高模型的泛化能力和鲁棒性。

4. 与 SWA 算法的测试准确率对比

在CIFAR-10数据集上，对ResNet-18模型分别使用RBWA、RBWA-E和SWA方法进行实验，对比模型在测试集上的准确率。实验中，批量大小为32，训练周期为50个epoch，RBWA、RBWA-E和SWA都从第10个epoch开始执行，保持其他实验设置完全一致。结果如图13所示。

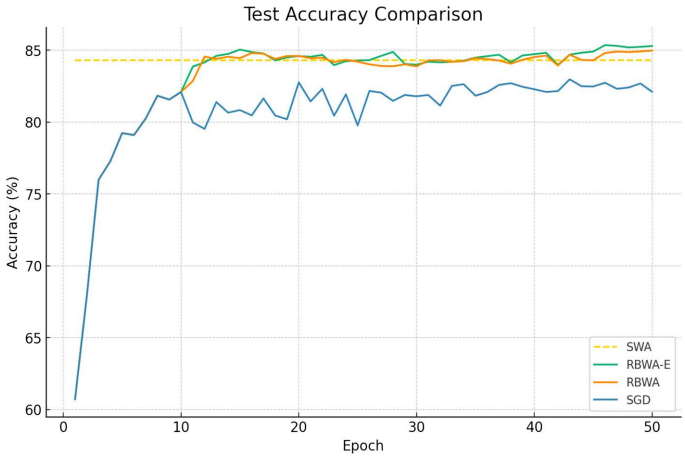

Figure 13. Comparison of test accuracy of RBWA, RBWA-E and SWA
图 13. RBWA、RBWA-E 和 SWA 的测试准确率对比

图 13 结果表明，相较于 SGD，RBWA 和 RBWA-E 在训练初期都能显著提高测试准确率，在训练结束时的测试准确率也明显高于 SWA。SWA 方法并不影响模型的训练过程，而 RBWA 和 RBWA-E 方法能显著提升损失函数收敛速度并增强模型更新过程的鲁棒性。因此，RBWA 和 RBWA-E 比传统的 SGD 和 SWA 方法，对模型的泛化能力和鲁棒性的影响更显著。

5. 总结

本文针对随机权重平均算法和循环随机权重平均优化算法的局限，提出了基于梯度下降的循环批量权重平均优化算法。该算法通过在每个训练周期内平均各个批次之间的权重并更新，有效减少由于小批量数据随机性引入的梯度噪声，从而使得梯度下降过程更加平稳。实验结果表明，与传统方法相比，RBWA 可以显著加快模型的训练过程，对损失函数的收敛速度有显著提升并且收敛到损失更小、方差更低的最优解。此外，在测试集上，使用 RBWA 方法的模型表现出更高的准确率，这表明该算法能够提高模型的泛化能力。RBWA 方法没有明显增加额外的计算开销，有利于集成到现有的优化框架中，提高了其在不同模型中的适用性。

虽然实验结果表明 RBWA 的效果优于 SGD 和 SWA，但是我们仍然缺乏理论支持，并且可能不总是适用于所有的深度神经网络任务。因此，未来的研究工作需要进行更多的理论和算法研究，并对更多不同类型和规模的深度神经网络任务进行广泛的训练和实验评估。这不仅能够帮助评估本文算法的适用性和局限性，也能有助于揭示算法在不同任务和数据集上的性能表现差异。此外，通过深入探索算法在各类任务中的应用效果，可以进一步优化和调整算法的设计，以适应更广泛的应用场景。

参考文献

[1] 谢正, 李浩, 宋伊萍, 等. 从 AIGC 到 AIGA, 智能新赛道: 决策大模型[J/OL]. 科学观察, 2024: 1-24. http://kns.cnki.net/kcms/detail/11.5469.N.20240412.1839.002.html, 2024-04-17.

[2] Robbins, H. and Monro, S. (1951) A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, **22**, 400-407. https://doi.org/10.1214/aoms/1177729586

[3] Roux, N., Schmidt, M. and Bach, F. (2012) A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets. *Proceedings of the* 25*th International Conference on Neural Information Processing Systems*, Lake Tahoe Nevada, 3-6 December 2012, 2663-2671.

[4] Johnson, R. and Zhang, T. (2013) Accelerating Stochastic Gradient Descent Using Predictive Variance Reduction. *Advances in Neural Information Processing Systems*, **26**, 315-323.

[5] Shalev-Shwartz, S. and Zhang, T. (2013) Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization. *Journal of Machine Learning Research*, **14**, 567-599.

[6] Nguyen, L.M., Liu, J., Scheinberg, K., *et al*. (2017) SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient. *Proceedings of the* 34*th International Conference on Machine Learning*, Sydney, 6-11 August 2017, 2613-2621.

[7] Konečný, J., Liu, J., Richtárik, P., *et al*. (2015) Mini-Batch Semi-Stochastic Gradient Descent in the Proximal Setting. *IEEE Journal of Selected Topics in Signal Processing*, **10**, 242-255. https://doi.org/10.1109/JSTSP.2015.2505682

[8] Beznosikov, A., Gorbunov, E., Berard, H. and Loizou, N. (2023) Stochastic Gradient Descent-Ascent: Unified Theory and New Efficient Methods. arXiv: 2202.07262.

[9] 王贝伦. 机器学习[M]. 南京: 东南大学出版社, 2021.

[10] Mignacco, F. and Urbani, P. (2022) The Effective Noise of Stochastic Gradient Descent. *Journal of Statistical Mechanics*: *Theory and Experiment*, No. 8, Article ID: 083405. https://doi.org/10.1088/1742-5468/ac841d

[11] Smith, S., Elsen, E. and De, S. (2020) On the Generalization Benefit of Noise in Stochastic Gradient Descent. arXiv: 2006.15081.

[12] Wojtowytsch, S. (2023) Stochastic Gradient Descent with Noise of Machine Learning Type Part I: Discrete Time Analysis. *Journal of Nonlinear Science*, **33**, Article No. 45. https://doi.org/10.1007/s00332-023-09903-3

[13] Izmailov, P., Podoprikhin, D., Garipov, T., *et al*. (2018) Averaging Weights Leads to Wider Optima and Better Generalization. arXiv: 1803.05407.

[14] Jain, P., Kakade, S.M., Kidambi, R., *et al.* (2018) Parallelizing Stochastic Gradient Descent for Least Squares Regression: Mini-Batching, Averaging, and Model Misspecification. *Journal of Machine Learning Research*, **18**, 1-42.

[15] Crowder, S.V. and Hamilton, M.D. (1992) An EWMA for Monitoring a Process Standard Deviation. *Journal of Quality Technology*, **24**, 12-21. https://doi.org/10.1080/00224065.1992.11979369