文章编号:1007－6735(2023)02－0112－08　DOI: 10.13255/j.cnki.jusst.20230106004

基于自适应动量更新策略的 Adams 算法

李满园，　罗　飞，　顾春华，　罗勇军，　丁炜超
（华东理工大学 信息科学与工程学院，上海 200237）

摘要： Adam 算法是目前最常用的优化算法之一，但其面临学习率震荡导致模型不收敛问题，其改进算法 AMSGrad 也存在梯度递减导致的二阶动量失效问题。针对上述问题，提出了基于自适应动量更新策略的 Adams 算法。首先，通过为一阶动量和二阶动量引入自适应更新参数，并在最后的参数更新期间采用较小的一阶动量更新参数，构建了一种自适应的动量更新策略。其次，基于该更新策略，提出了一种能够快速收敛的 Adams 算法。最后，通过理论分析证明了 Adams 算法的收敛性。基于文本分类和图像分类的对比实验表明，相比于 Adam 和 AMSGrad 算法，Adams 收敛速度更快、训练结果更好，且具有优秀的泛化能力；消融实验证明了 Adams 算法自适应动量更新策略的有效性。

关键词： 优化算法；自适应动量更新策略；一阶动量；二阶动量
中图分类号： TP 301.6　　**文献标志码：** A

Adams algorithm based on adaptive momentum update strategy

LI Manyuan,　LUO Fei,　GU Chunhua,　LUO Yongjun,　DING Weichao
(*School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China*)

Abstract: Adam is one of the most commonly used optimization algorithms at present, but it is faced with the problem that the model does not converge due to the vibration of learning rate, and its improved algorithm AMSGrad also has the problem of invalid second-order momentum caused by gradient decline. To solve the above problems, Adams optimization algorithms based on adaptive momentum update strategy was proposed. Firstly, an adaptive momentum update strategy was constructed by introducing adaptive update parameters for the first order momentum and the second order momentum, and a smaller first order momentum update parameters was used during the final parameter update phase. Secondly, based on this update strategy, a fast convergence algorithm named Adams was proposed. Finally, the convergence of Adams algorithm was analyzed theoretically. The comparative experiments based on text classification and image classification show that Adams has faster convergence speed, better training results, and more excellent generalization ability than Adam and AMSGrad. The ablation experiment proves the effectiveness of Adams adaptive momentum updating

收稿日期: 2023−01−06
基金项目: 国家自然科学基金资助项目(62276097)；上海市自然科学基金资助项目 (22ZR1416500)；上海市青年科技英才扬帆计划(20YF1410900)；上海市“科技创新行动计划”(21002411000)
第一作者: 李满园(1996−)，男，硕士研究生．研究方向：推荐算法．E-mail：manyuan1222@163.com
通信作者: 罗　飞(1973−)，男，副教授．研究方向：分布式计算、云计算、强化学习．E-mail：luof@ecust.edu.cn

strategy.

Keywords: *optimization algorithm*; *adaptive momentum update strategy*; *first-order momentum*; *second-order momentum*

近年来，人工神经网络发展迅猛，已经被广泛应用于人脸识别[1]、目标检测[2]、文本翻译[3]、数据预测[4]及医疗[5]等领域。相较于传统的机器学习算法，神经网络能够更好地利用海量的数据[6]。为了达到更好的效果，神经网络需要进行大量的训练，并通过优化算法来不断地减少损失、优化模型。Adam 算法[7]是一种简单且高效的自适应学习率优化算法，可以在训练过程中根据历史梯度自适应调整学习率来获取更好的训练结果[8]。Adam 算法易于使用，Yi 等[9]的实验证明，其仅使用默认参数就可以获得较好的训练结果，而且，它的计算效率高、占用内存小，适合解决大规模数据和多参数优化问题[10]。因为上述优点，Adam 算法目前已经成为神经网络训练中使用最多的优化算法之一。但是，在 Adam 算法中，为了避免学习率过度衰减导致训练提前结束[11]，采用固定时间窗口内历史梯度的累积作为二阶动量。在多次迭代过程中，若训练样本间梯度差过大，会出现二阶动量波动引起的学习率震荡，最终导致模型不收敛[12]。此外，Adam 算法还存在泛化能力差、可能因过拟合而错过全局最优解[13]等问题。

为了解决上述问题，并进一步提升 Adam 算法的性能，不少研究者提出了自己的改进方法。Dozat[14]利用 NAG（nesterov accelerated gradient）对 Adam 算法的一阶动量进行更新，从而提升收敛速度。文献 [15-16] 指出几乎所有深度学习库中，对于包括 Adam 算法的权重衰减方法的实现都存在错误，并在 Adamw 算法中进行修正。Reddi 等[12]提出 AMSGrad 算法，通过改进二阶动量的迭代方式来避免学习率的震荡，从而解决模型不收敛问题。Lazy Adam 算法[17]是 Adam 算法的一种常用的变体，通过引入懒惰梯度更新机制实现对稀疏梯度问题的高效处理，更适用于自然语言处理等梯度稀疏任务。Keskar 等[18]提出在训练的后期，从 Adam 算法切换到 SGD（stochastic gradient descent）算法[19]可以使模型获得更好的泛化性能。

在上述对于 Adam 算法的改进中，AMSGrad 算法解决了 Adam 算法中存在的模型不收敛问题。但是，当训练中多次迭代的梯度保持不变或整体呈递减趋势，AMSGrad 算法的二阶动量会无法正常更新而导致自适应学习率失效。针对这一问题，本文分析了自适应学习率失效的原因，并提出基于自适应动量更新策略的 Adams 算法。本文的主要研究工作为：a. 实现二阶动量自适应更新，在解决模型不收敛的同时，避免自适应学习率失效问题；b. 实现一阶动量参数在训练过程中的自适应增加，获取更快的收敛速度；c. 在最后的参数更新阶段中，采用较小的一阶动量更新参数，使模型完成更加精细的收敛。

1　Adams 算法设计

1.1　自适应学习率失效问题

为了避免 Adam 算法中因为二阶动量波动导致的模型不收敛问题，AMSGrad 算法优化了二阶动量的更新公式，优化后的公式为

$$v_t \leftarrow \max(\beta_2 v_{t-1} + (1-\beta_2) g_t^2,\ v_{t-1}) \qquad (1)$$

式中：g_t 为梯度；v_t 为梯度的二阶原点矩的有偏估计，即二阶动量；β_2 为二阶动量的指数移动加权衰减率，即二阶动量更新系数。

由式（1）可知，AMSGrad 算法取迭代前后二阶动量的最大值作为新的二阶动量的值。这一策略可以防止二阶动量在整个训练的过程中出现波动，通过避免学习率震荡来解决模型不收敛问题。但是，若训练过程中梯度整体呈下降趋势，二阶动量则会始终保持不变，无法正常更新，导致算法的自适应学习率失效。

1.2　自适应动量更新策略

一方面为了在解决 Adam 算法不收敛问题时避免出现自适应学习率失效问题，另一方面为了加速模型的收敛，针对一阶动量和二阶动量，提出了自适应动量更新策略。自适应动量更新策略的设计主要考虑以下几个方面的问题：a. 通过为二阶动量引入恒大于 1 的更新参数，令其在整个训练过程中严格单调递增，避免了学习率的震荡导致的模型不收敛问题。同时，参数不能过大，否

则会因二阶动量过度累积导致训练提前结束。b. 令一阶动量更新参数随着训练进行逐渐变大，并在最后的参数更新阶段将其设置为较小的值。自适应动量更新策略分为二阶动量改进和一阶动量改进这两部分。

首先，综合考虑Adam算法中的模型不收敛问题与AMSGrad算法中的自适应学习率失效问题，为二阶动量引入自适应更新参数μ，其定义为

$$\mu \leftarrow 1+\alpha/v_{t-1} \tag{2}$$

式中，α为算法的预设学习率。

因为，学习率α恒为正，且二阶动量v_t恒为正，所以，μ恒大于1。μ可以使二阶动量在训练中严格单调递增，同时避免学习率震荡和自适应学习率失效两个问题。此外，μ与算法的实际学习率线性正相关，会在训练中自适应减小，因此，不会出现梯度过度累积导致训练提前结束的问题。基于自适应更新参数μ构建的自适应二阶动量更新方法，如式(3)所示。

$$v_t \leftarrow \max(\beta_2 v_{t-1}+(1-\beta_2)g_t^2, \mu v_{t-1}) \tag{3}$$

其次，Sutskever等[20]的研究表明：一方面，令一阶动量的更新参数β_1在训练过程中缓慢地增大，可以使模型收敛更快，从而达到Hessian-Free优化[21]的效果；另一方面，在最后的参数更新阶段采用较小的一阶动量更新参数可以获得更加精细的收敛结果。因此，令β_1在每次迭代完成后乘以μ来缓慢增大，以获取更快的收敛速度。同时，采用文献[20]的策略，即在最后1000次参数更新期间将β_1设置为0.9，以获取更好的收敛结果，并设置2个消融实验分别证明这两点改进的有效性。应用这一改动后，一阶动量更新公式为

$$m_t \leftarrow \mu^{t-1}\beta_1 m_{t-1}+(1-\mu^{t-1}\beta_1)g_t \tag{4}$$

式中：m_t为梯度的一阶原点矩的有偏估计，即一阶动量；β_1为一阶动量的指数移动加权衰减率，即一阶动量更新参数。

1.3 Adams算法

算法1简要描述基于自适应动量更新策略的Adams优化算法的执行流程。

算法1 Adams算法。

Input: β_1, β_2, α Output: θ_t

initialize $m_0 \leftarrow 0$, $v_0 \leftarrow 0$, $t \leftarrow 1$, $\mu \leftarrow 1$, random(θ_0)

(a) while θ_t non-convergence, then

(b) $g_t \leftarrow \nabla_\theta f(\theta_{t-1})$

(c) $m_t \leftarrow \mu^{t-1}\beta_1 m_{t-1}+(1-\mu^{t-1}\beta_1)g_t$

(d) $v_t \leftarrow \max(\beta_2 v_{t-1}+(1-\beta_2)g_t^2, \mu v_{t-1})$

(e) $M_t \leftarrow m_t/(1-\beta_1^t)$

(f) $V_t \leftarrow v_t/(1-\beta_2^t)$

(g) $\mu \leftarrow 1+\alpha/(v_{t-1}+\varepsilon)+1$

(h) $\theta_t \leftarrow \theta_{t-1}-\alpha M_{t-1}/(\sqrt{V_{t-1}}+\varepsilon)$

(i) $t \leftarrow t+1$

在算法1中，第2行计算出当前梯度g_t后，第3，4行分别计算g_t的一阶原点矩估计m_t和二阶原点矩估计v_t，即一阶动量和二阶动量。在迭代初期，计算出的m_t和v_t都是有偏的，因此，分别在第5，6行对其进行偏差修正。迭代后期，对梯度的矩估计逐渐变为无偏估计，修正强度也逐渐降为1。算法1中第7行得到自适应更新参数μ，用于m_t和v_t的自适应更新。最后，在第8行完成模型参数的更新。此外，在最后1000次参数更新中将β_1设置为0.9并未在算法1中详细写出。

2 Adams算法收敛性证明

为了证明Adams算法的收敛性，需要使用算法收敛性评判的标准定义，以及一些收敛性分析的常用假设，在此基础上，对Adams算法的收敛性进行理论分析。

定义1 对于任意给定的未知凸代价函数系列$f_1(\theta)$，$f_2(\theta)$，$\cdots$，$f_T(\theta)$，算法在迭代中给出预测参数模型θ_t。用任意给定凸代价函数f_t对其进行评估，并使用收敛指标$R(T)$来对算法进行整体性的评估。当T趋于无穷大时，$\lim(R(T)/T)=0$，则认为此算法收敛。$R(T)$定义为

$$R(T)=\sum_{i=1}^{t}[f_i(\theta_i)-\min_\theta f_i(\theta)] \tag{5}$$

假设1 算法的实际学习率序列$\{\eta_t\}$是一个非增序列，且满足以下条件：

$$\sum_{t=1}^{\infty}\eta_t=\infty\,,\ \sum_{t=1}^{\infty}\eta_t^2<\infty \tag{6}$$

假设2 对于$\forall t\in N^+$，凸优化函数f_t的梯度有界，即存在常量G，G_∞，使梯度的二阶范数和无穷范数满足以下条件：

$$\|\nabla f_t(\theta)\|_2 \leqslant G,\quad \|\nabla f_t(\theta)\|_\infty \leqslant G_\infty \tag{7}$$

假设3 Adams算法预测的任意参数θ_t之间，差值有界，即参数θ_t的二阶范数和无穷范数满足以下条件：

$$\|\theta_n-\theta_m\|_2 \leqslant G,\ \|\theta_n-\theta_m\|_\infty \leqslant G_\infty \tag{8}$$

定理 1　对于$\forall T \geqslant 1$，Adams 算法对应的 $R(T)$ 有上界，即 Adams 是收敛的。

证明　在 Adams 算法中，一阶动量更新参数 β_1 会在训练中以 μ 为更新系数缓慢增大，记作 $\{\mu^0\beta_1, \mu^1\beta_1, \cdots, \mu^{t-1}\beta_1\}$。此外，为了证明过程的简洁，引入常量 G 和 D，在忽略极小值 ε 后，有

$$\theta_{t+1} = \theta_t - \gamma_t \frac{\mu^{t-1}\beta_1 M_{t-1} + (1-\mu^{t-1}\beta_1) g_t}{\sqrt{V_t}} \tag{9}$$

其中，

$$\gamma_t = \alpha / (1 - \prod_{s=1}^{t} \mu^{s-1}\beta_1) \tag{10}$$

在式 (9) 两边同时减去θ^*后平方，并在展开之后将一阶动量更新公式（4）代入，整理可得

$$g_t(\theta_t - \theta^*) = \frac{\sqrt{V_t}\left[(\theta_t-\theta^*)^2 - (\theta_{t+1}-\theta^*)^2\right]}{2\gamma_t(1-\mu^{t-1}\beta_1)} - \frac{\mu^{t-1}\beta_1}{1-\mu^{t-1}\beta_1} M_t(\theta_t - \theta^*) + \frac{\gamma_t}{2(1-\mu^{t-1}\beta_1)} \frac{(M_t)^2}{\sqrt{V_t}} \tag{11}$$

其中，

$$\theta^* = \arg\min_{\theta} \sum_{t=1}^{T} f_t(\theta) \tag{12}$$

在假设 2 和假设 3 同时成立的基础上，令 $A = \sum_{t=1}^{T} \sqrt{V_t}\left[(\theta_t-\theta^*)^2 - (\theta_{t+1}-\theta^*)^2\right] \Big/ \left(2\gamma_t(1-\mu^{t-1}\beta_1)\right)$，通过放缩可得 A 的上界：

$$A = \sum_{t=1}^{T} \frac{\sqrt{V_t}\left[(\theta_t-\theta^*)^2 - (\theta_{t+1}-\theta^*)^2\right]}{2\alpha_t(1-\mu^{t-1}\beta_1)} \cdot \left(1 - \prod_{s=1}^{t}\mu^{s-1}\beta_1\right) \leqslant \frac{GD^2}{2\alpha(1-\mu^{t-1}\beta_1)} \tag{13}$$

令 $B = \sum_{t=1}^{T} -\frac{\mu^{t-1}\beta_1}{1-\mu^{t-1}\beta_1} M_t(\theta_t - \theta^*)$，在假设 2 和假设 3 中有界条件成立的基础上，通过放缩可得上界：

$$B \leqslant \sum_{t=1}^{T} GD \frac{\mu^{t-1}\beta_1}{1-\mu^{t-1}\beta_1} = GD \sum_{t=1}^{T} \frac{\mu^{t-1}\beta_1}{1-\mu^{t-1}\beta_1} \tag{14}$$

令 $C = \frac{\gamma_t}{2(1-\mu^{t-1}\beta_1)} \frac{M_t^2}{\sqrt{V_t}}$，在假设 2 和假设 3 中有界条件成立的基础上，通过放缩可得 C 的上界：

$$C \leqslant G \sum_{t=1}^{T} \frac{\gamma_t}{2(1-\mu^{t-1}\beta_1)} \cdot \sum_{s=1}^{t} \frac{(1-\mu^{s-1}\beta_1)^2 \left(\prod_{r=s+1}^{t} \mu^{r-1}\beta_1\right)^2}{(1-\beta_2)\beta_2^{t-s}} \tag{15}$$

将 A，B，C 三者上界相加可得 $R(T)$，即

$$A + B + C = R(T) = \sum_{t=1}^{T} g_t(\theta_t - \theta^*) \tag{16}$$

至此，证得 $R(T)$ 存在上界。根据定义 1 可得结论：Adams 算法收敛。

3　实验设计与结果分析

3.1　实验环境

基于深度学习框架 PyTorch 实现了基于自适应动量更新策略的 Adams 算法，实验环境中的多个主要工具包的具体版本如表 1 所示。

表 1　实验工具包版本

Tab.1　Toolkit version in experiments

工具包	版本
python	3.8.15
torch	1.10.0+cu113
torchvision	0.11.1

在 reuters，MNIST，fashion MNIST，CIFAR10，CIFAR100 共 5 个公共数据集上，通过文本分类与图像分类实验，测试 Adams 算法的性能。其中，reuters 是来自路透社的短新闻文本数据集；MNIST 是手写数字的灰度图像数据集；fashion MNIST 是包含不同种类商品的灰度图像数据集；CIFAR10，CIFAR100 是包含不同种类物品的彩色图像数据集。5 个数据集的数据量、训练集与测试集划分、数据特征如表 2 所示。

3.2　算法对比实验与分析

Adams 算法通过自适应动量更新策略解决 Adam 算法不收敛问题的同时，避免了 AMSGrad 算法中的自适应学习率失效问题。因此，选取 Adam 算法和 AMSGrad 算法进行对比实验。上述提及的算法均为自适应学习率算法，而 Wilson 等[16]的研究表明，自适应学习率算法得到模型的泛化能力通常不如带动量的 SGD 算法（SGD with momentum，简称 SGD），因此，将 SGD 算法也作为对比对象。考虑到不同算法在不同学习率、不同数据集的表现可能不同，算法均训练 20 轮，步长均设置为 64，学习率分别设置为 0.000 1，0.001，0.002，0.003，0.004，0.005，0.01，0.05，0.1，进行多次实验，并选取最佳结果作为实验结果。

Adams 算法的目的是帮助神经网络更快地训练出更好的模型。为了比较 Adams 算法在不同类

表 2 实验数据集

Tab.2 Experimental datasets

数据集	样本数	训练集	测试集	类别	数据特点
reuters	11228	9000	2228	46	文本数据,特征简单,类别稍多
MNIST	70000	60000	10000	10	灰度图像,特征复杂,类别少
fashion MNIST	70000	60000	10000	10	灰度图像,特征复杂,类别少
CIFAR10	60000	50000	10000	10	RGB图像,特征复杂,类别少
CIFAR100	60000	50000	10000	100	RGB图像,特征复杂,类别多

型的神经网络中的表现，也为了凸显对比实验的对比性，在不同的数据集上，将使用不同的神经网络进行训练。在 reuters，MNIST，fashion MNIST 这 3 个相对较为简单的数据集上进行实验时，使用仅包含一个线性层的全连接神经网络（fully connected neural network, FCNN）进行训练，避免因为训练过快而导致实验结果失去对比性；在 CIFAR10 和 CIFAR100 这 2 个相对复杂的数据集上进行实验时，采用包含 3 个卷积模块的卷积神经网络（convolutional neural networks, CNN），网络结构如图 1 所示。本实验中所有的神经网络的输出都通过 softmax 函数进行归一化处理，并使用交叉

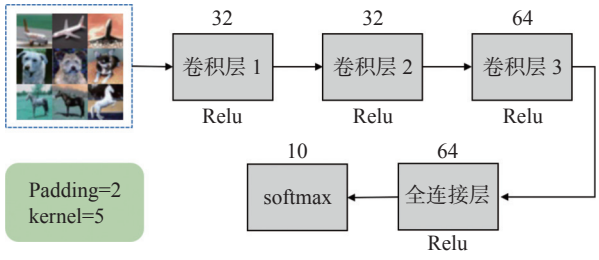


图 1 算法对比实验中的 CNN 模型

Fig. 1 CNN model in comparison experiments

熵计算损失来进行参数更新，具体训练结果如表 3 所示。Relu，softmax 为激活函数。

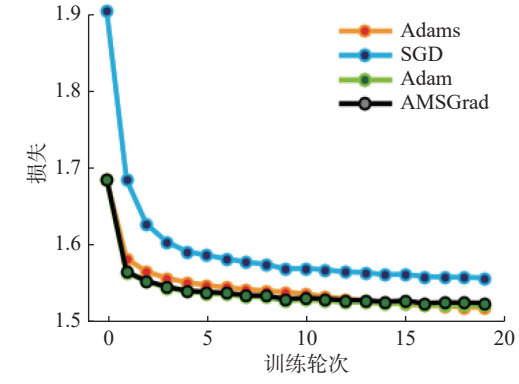
为了更为直观地对比 4 种算法，图 2 以 MNIST

表 3 算法对比实验结果

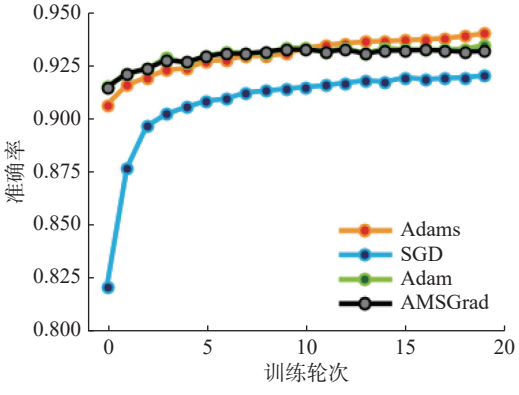
Tab.3 Results of algorithm comparison experiments

数据集	优化算法	学习率	损失	准确率/%
reuters	SGD	0.005	1.3312	70.72
	Adam	0.001	0.1093	95.54
	Adams	0.002	**0.1007**	**96.12**
	AMSGrad	0.001	0.1026	95.83
MNIST	SGD	0.100	1.5563	92.02
	Adam	0.002	1.5204	93.44
	Adams	0.001	**1.5179**	**94.01**
	AMSGrad	0.002	1.5237	93.18
fashion MNIST	SGD	0.100	1.6347	82.90
	Adam	0.001	1.5992	84.73
	Adams	0.002	**1.5825**	**85.74**
	AMSGrad	0.002	1.5916	85.04
CIFAR10	SGD	0.050	0.4281	70.40
	Adam	0.001	0.4137	73.15
	Adams	0.001	**0.3962**	**74.82**
	AMSGrad	0.001	0.4246	72.91
CIFAR100	SGD	0.050	1.5163	32.53
	Adam	0.001	1.8759	37.12
	Adams	0.001	**1.8603**	**38.27**
	AMSGrad	0.001	1.8732	38.10

数据集为例，给出了 4 种算法的收敛过程，各算法学习率的设置与表 3 中相同。从图 2(a)可以看出，在 MNIST 数据集上，SGD 的收敛速度最慢，另外 3 种算法的收敛速度相差不大，其中，Adams 收敛速度略快。由图 2(b)可知，Adams 算法在采用自适应动量更新策略后，得到了更为精细的收敛结果，明显优于 SGD，Adam，AMSGrad 这 3 种算法。



(a) MNIST 对比实验中各算法的损失



(b) MNIST 对比实验中各算法的准确率

图 2　MNIST 上的对比实验

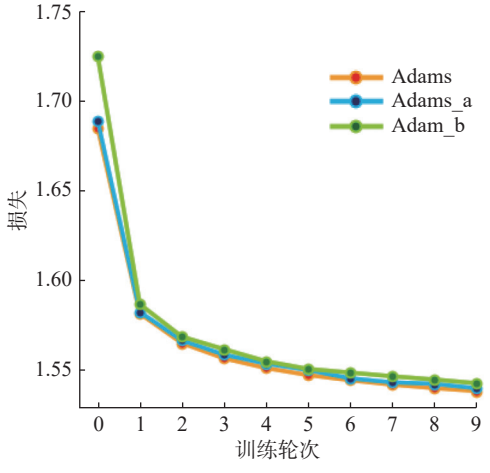
Fig.2　Comparison experiments on MNIST

在所有数据集上的实验结果如表 3 所示。从表 3 可以看出，Adams 算法在 5 个不同数据集上的对比实验中均获得了最高的分类准确率。并且在训练了 20 轮之后，Adams 均得到了最低的损失，这也说明其收敛速度更快。此外，Adams 算法在 FCNN，CNN 两类神经网络上都表现出了更好的性能；在文本分类任务 (reuters) 和图片分类任务 (MNIST，fashion MNIST，CIFAR10，CIFAR100) 上也都表现出较好的性能，证明了 Adams 算法具有优秀的泛化能力。

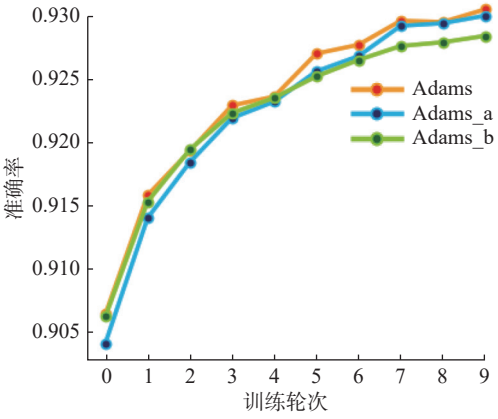
3.3　消融实验与分析

相比于 Adam 算法，Adams 算法在自适应动量更新策略中进行了 3 点改进，分别为：a. 实现二阶动量自适应更新；b. 实现一阶动量参数在训练过程中的自适应增加；c. 在最后的参数更新阶段采用较小的一阶动量更新参数。为了进一步探究前 2 点改进各自的作用，本文在 Adam 算法的基础上分别单独实现这 2 点改动，并分别记作 Adams_a 算法、Adams_b 算法，与 Adams 算法一起，在 MNIST 数据集上进行消融实验。MNIST 数据集内数据特征简单，如果采用比较复杂的网络进行训练，在数次迭代后 4 种优化算法都会达到较高的准确率。为了使实验结果更加具有可比性，在消融实验中仍然使用仅包含 1 个线性层的全连接神经网络，训练轮次为 10 次。消融实验的具体实验结果如图 3 所示。

由图 3(a)可知，在 Adams_a 算法中实现二阶动量自适应更新后，获得了与 Adams 算法相近的损失与准确率，但损失下降速度慢；由图 3(b)可知，在 Adams_b 算法中实现一阶动量参数的自适



(a) MNIST 对比实验中各算法的损失



(b) MNIST 对比实验中各算法的准确率

图 3　消融实验中多种算法对比

Fig. 3　Comparison of various algorithms in ablation experiments

应增加后，得到了更快的损失下降速度，但最终模型损失较高，且准确率较低。这些现象的出现，也进一步证明了：a. 二阶动量的自适应更新可以使 Adams 算法获得更好的收敛结果；b. 一阶动量更新参数在训练中自适应增加，并在最后 1 000 次参数更新中设置为 0.9，可以使 Adams 算法获得更快的收敛速度。

为了验证 Adams 算法中第 3 点改进的有效性，将仅应用前 2 点改进的算法记作 Adams_c 算法，并与 Adams 算法一起在 MNIST 上进行消融实验。此实验是为了验证最终的收敛结果是否更加精细，因此，采用早停法，当 Adams_c 算法出现训练次数增多而模型表现变差的情况时停止训练。Adams 算法训练相同次数。除此之外，实验采用与算法对比实验相同的参数设置，实验结果如表 4 所示。从表 4 中数据可知，这一改动可以略微提升模型的性能。

表 4 消融实验结果

Tab.4 Results of ablation experiments

算法	学习率	损失	正确率/%
Adams	0.001	**1.5179**	**94.01**
Adams_c	0.001	1.5263	93.39

4 结束语

针对 Adam 算法中因为学习率震荡导致的模型不收敛问题，以及 AMSGrad 算法中自适应学习率失效的问题，提出了基于自适应动量更新策略的 Adams 算法。在 Adams 算法中，一方面通过为二阶动量引入自适应更新参数来避免不收敛问题和自适应学习率失效问题；另一方面使一阶动量在训练中自适应增大来获取更快的收敛速度，并在最后 1 000 次参数更新期间采用较小的一阶动量更新参数来完成更加精细的收敛。

为了对 Adams 算法进行性能评估，在 reuters，MNIST，fashion MNIST，CIFAR10，CIFAR100 共 5 个数据集上，与 SGD，Adam，AMSGrad 算法进行对比。对比实验证明，Adams 算法相比其他 3 种优化算法，收敛速度更快、收敛结果更好。为了进一步验证了 Adams 算法中 2 点改动的有效性，在 MNIST 数据集上进行消融实验。消融实验证明，二阶动量的自适应更新可以使算法获得更好的收敛结果；一阶动量更新参数的自适应增大可以使算法获得更快的收敛速度；在最后的参数更新阶段中，采用较小的一阶动量更新参数可以使模型获得更加精细的收敛结果。

参考文献：

[1] WANG M, DENG W H. Deep face recognition: a survey[J]. Neurocomputing, 2021, 429: 215–244.

[2] ZHAO Z Q, ZHENG P, XU S T, et al. Object detection with deep learning: a review[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(11): 3212–3232.

[3] SHEN X P, QIN R J. Searching and learning English translation long text information based on heterogeneous multiprocessors and data mining[J]. Microprocessors and Microsystems, 2021, 82: 103895.

[4] 张兆旭, 刘成忠. 基于 IFOA-GRNN 的电力系统短期负荷预测 [J]. 能源研究与信息, 2020, 36(3): 162–166.

[5] 秦晓飞, 郑超阳, 陈浩胜, 等. 基于 U 型卷积网络的视网膜血管分割方法 [J]. 光学仪器, 2021, 43(2): 24–30.

[6] SEWAK M, SAHAY S K, RATHORE H. Comparison of deep learning and the classical machine learning algorithm for the malware detection[C]//19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. Busan: IEEE, 2018: 293–296.

[7] KINGMA D P, BA J. Adam: A method for stochastic optimization[C]//3rd International Conference on Learning Representations. San Diego: ICLR, 2015: 1–15.

[8] AITCHISON L. Bayesian filtering unifies adaptive and non-adaptive neural network optimization methods[C]// Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc. , 2020: 18173–18182.

[9] YI D, AHN J, JI S. An effective optimization method for machine learning based on ADAM[J]. Applied Sciences, 2020, 10(3): 1073–1093.

[10] 周扬帆. 面向深度学习的随机梯度优化算法研究 [D]. 洛阳: 河南科技大学, 2020.

[11] SINGH B, DE S, ZHANG Y M Z, et al. Layer-specific adaptive learning rates for deep networks[C]//IEEE 14th International Conference on Machine Learning and Applications. Miami: IEEE, 2015: 364–368.

[12] REDDI S J, KALE S, KUMAR S. On the convergence of Adam and beyond[C]//The 6th International Conference on Learning Representations. Vancouver: ICLR, 2018: 1–23.

[13] WILSON A C, ROELOFS R, STERN M, et al. The marginal value of adaptive gradient methods in

machine learning[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc. , 2017: 4151–4161.

[14] DOZAT T. Incorporating nesterov momentum into Adam[C]//The 4th International Conference on Learning Representations. San Juan: Open Review, 2016: 1–4.

[15] ILYA L, FRANK H. Fixing weight decay regularization in Adam[C]//The 6th International Conference on Learning Representations. Vancouver: Open Review, 2018: 1–14.

[16] LOSHCHILOV I, HUTTER F. Decoupled weight decay regularization[C]//The 7th International Conference on Learning Representation. New Orleans: ICLR, 2019: 1–4.

[17] QURESHI M N, UMAR M S, SHAHAB S. A transfer-learning-based novel convolution neural network for melanoma classification[J]. Computers, 2022, 11(5): 64.

[18] KESKAR N S, SOCHER R. Improving generalization performance by switching from Adam to SGD[DB/OL]. [2017-12-20]. https://arXiv preprint arXiv: 1712.07628.

[19] ROBBINS H, MONRO S. A stochastic approximation method[J]. The Annals of Mathematical Statistics, 1951, 22(3): 400–407.

[20] SUTSKEVER I, MARTENS J, DAHL G, et al. On the importance of initialization and momentum in deep learning[C]//Proceedings of the 30th International Conference on International Conference on Machine Learning. Atlanta: JMLR. org, 2013: 1139–1147.

[21] MARTENS J. Deep learning via hessian-free optimization [C]//Proceedings of the 27th International Conference on Machine Learning. Haifa: ICML, 2010: 735–742.

（编辑：石　瑛）

[上接第 111 页]

[16] LI R F, CHEN H, FENG F X, et al. Dual graph convolutional networks for aspect-based sentiment analysis[C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. Stroudsburg: ACL, 2021: 6319–6329.

[17] WEI X D, HUANG H, MA L X, et al. Recurrent graph neural networks for text classification[C]//Proceedings of the IEEE 11th International Conference on Software Engineering and Service Science. New York: IEEE, 2020: 91–97.

[18] 闫佳丹, 贾彩燕. 基于双图神经网络信息融合的文本分类方法 [J]. 计算机科学, 2022, 49(8): 230–236.

[19] 范国凤, 刘璟, 姚绍文, 等. 基于语义依存分析的图网络文本分类模型 [J]. 计算机应用研究, 2020, 37(12): 3594–3598.

[20] 邵党国, 张潮, 黄初升, 等. 结合 ONLSTM-GCN 和注意力机制的中文评论分类模型 [J]. 小型微型计算机系统, 2021, 42(7): 1377–1381.

[21] QI P, DOZAT T, ZHANG Y H, et al. Universal dependency parsing from scratch[C]//Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Brussels: Association for Computational Linguistics, 2018: 160–170.

[22] 闫育铭, 李峰, 罗德名, 等. 基于深度迁移学习的糖尿病视网膜病变的检测 [J]. 光学仪器, 2020, 42(5): 33–42.

（编辑：石　瑛）