基于 Nesterov 加速的改进自适应优化算法

钱 振[1],李德权[2]
1. 安徽理工大学 数学与大数据学院,安徽 淮南 232001
2. 安徽理工大学 人工智能学院,安徽 淮南 232001

摘 要:目的 针对传统优化算法在训练深度学习模型时,由于模型参数量不断增大,网络层数不断加深所产生的训练效率较低的问题,提出一种基于 Nesterov 加速的 Nadabelief 优化算法,以提高模型的训练效率。**方法** 首先采取 Adabelief 算法代替 Adam 算法,缓解了算法的泛化性问题;接着从一阶矩经典动量项的角度出发,在 Adabelief 算法的基础上引入了 Nesterov 动量加速机制,在梯度更新时不仅考虑当前时刻的梯度,还借助于历史累积梯度来修正梯度的更新幅度,进一步提升了算法的效率;最后根据理论分析证明得到算法的遗憾界,确保了算法的收敛性。**结果** 为了验证算法的性能,在凸情况下进行了 Logistic 回归实验,在非凸情况下进行了图像分类和语言建模实验,通过与 Adam、Adabelief 等算法的比较,验证了 Nadabelief 算法的优越性。通过在不同初始学习率下对算法进行测试,验证了算法良好的鲁棒性。**结论** 实验表明:所提出的算法在保持原有 Adabelief 算法泛化能力的同时兼具更好的收敛精度,在训练深度学习模型时效率得到了进一步提高。

关键词:自适应算法;Nesterov 动量加速;深度学习;图像识别;语言建模
中图分类号:TP18 **文献标识码:**A doi:10.16055/j.issn.1672-058X.2025.0003.006

An Improved Adaptive Optimization Algorithm Based on Nesterov Acceleration

QIAN Zhen[1], LI Dequan[2]
1. School of Mathematics and Big Data, Anhui University of Science and Technology, Anhui Huainan 232001, China
2. School of Artificial Intelligence, Anhui University of Science and Technology, Anhui Huainan 232001, China

Abstract: Objective Traditional optimization algorithms exhibit lower training efficiency when training deep learning models due to increasing model parameters and deeper network layers. To address this issue, a Nadabelief optimization algorithm based on Nesterov acceleration was proposed to improve the efficiency of model training. **Methods** Firstly, the Adabelief algorithm was employed in place of the Adam algorithm to mitigate the generalization problem. Subsequently, from the perspective of the first-order moment classical momentum term, the Nesterov momentum acceleration mechanism was incorporated into the Adabelief algorithm. During gradient updates, not only the gradient at the current moment was considered, but the historical cumulative gradient was also utilized to adjust the magnitude of gradient updates, so as to further improve the convergence of the algorithm. Finally, the regret bound of the algorithm was obtained based on theoretical analysis to ensure the convergence of the algorithm. **Results** To verify the performance of the algorithm, Logistic regression experiments were conducted in the convex scenario, while image classification and language modeling experiments were carried out in the non-convex scenario. Comparisons with algorithms such as Adam and Adabelief demonstrated the superiority of the Nadabelief algorithm. Additionally, the algorithm's robustness was confirmed by testing it at various initial learning rates. **Conclusion** The experiments demonstrate that the proposed algorithm not only

收稿日期:2023-10-07 **修回日期:**2023-11-23 **文章编号:**1672-058X(2025)03-0044-08
基金项目:安徽省学术和技术带头人及后备人选项目(2019H211).
作者简介:钱振(1999—),男,安徽黄山人,硕士研究生,从事算法优化、深度学习研究.
通信作者:李德权(1973—),男,安徽肥东人,博士,教授,博士生导师,从事人工智能算法研究. Email:leedqcpp@126.com.
引用格式:钱振,李德权. 基于 Nesterov 加速的改进自适应优化算法[J]. 重庆工商大学学报(自然科学版),2025,42(3):44-51.
QIAN Zhen, LI Dequan. An improved adaptive optimization algorithm based on Nesterov acceleration[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2025, 42(3): 44-51.

maintains the generalization capability of the original Adabelief algorithm but also achieves better convergence accuracy. The proposed algorithm further improves the efficiency when training deep learning models.

Keywords: adaptive algorithms; Nesterov momentum acceleration; deep learning; image recognition; language modeling

1　引　言

随着计算机信息技术的快速发展,深度学习技术日益成熟,被广泛应用于语义分割、语言建模和机器人路径规划等实际问题。与此同时,随着卷积神经网络等[1]的不断深入应用,模型层数不断加深,参数量不断扩大,选择一种合适的优化算法变得至关重要。随机梯度下降法[2]是深度学习中常用的优化算法,可是该算法却存在收敛速度慢、算法无法保证收敛到全局最优等问题。

为此,学者们提出了加速梯度下降法[3]与自适应算法等优化算法来缓解这些问题。Sutskever[4]在 2013 年提出 Nesterov 动量加速机制以提高收敛速度;Kingma[5]提出 Adam 算法,Duchi[6]设计 Adagrad 算法,为每个参数设计独立的自适应学习率,训练效率进一步提高;Tang[7]针对大型模型开发出了 1-bit Adam 算法,这些算法都提高了模型训练效率。

作为自适应优化算法中常使用的算法之一,Adam 算法可以在不同情况下自适应地调整参数的学习率,这使得它的收敛速度比 SGD 算法更快。但是 Reddi 等[8]指出,Adam 算法有时即使在简单的凸设置下,也会出现不收敛的情况,此外它的泛化能力也较弱;Wilson[9]提到 Adam 算法在训练后期会产生不稳定和极端学习率现象,影响算法的收敛精度。

针对 Adam 算法存在的缺陷,学者们提出了一系列 Adam 算法的变体进行改进:Reddi 等[8]设计了 Amsgrad 算法,利用梯度值中的最大值代替梯度平均值来更新学习率,从而降低了过高的学习率,以确保算法收敛;Reyad 等[10]通过将 Adam 与 Amsgrad 相结合,建立一种混合机制以提升收敛速度。上述两种算法都采取最大值方法,只考虑了学习率过大时的情况,学习率过小时的情况则没有考虑;Lu[11]针对算法中需要手动调参的问题,提出一种对超参数不敏感的算法,因此不需要和其他自适应算法一样频繁地手动调整超参数;Duman[12]和 Agarwal[13]在花卉数据集与动物数据集上对不同优化算法在图像识别方面的性能做了对比,衡量了不同优化算法的性能,但只是基于算法的简单应用,对于算法本身并未提供改进,算法本身的振荡等问题也并未改善。

Yang 和 Fang 等[14-15]通过对学习率进行动态裁剪,使学习率向量中的每个元素都被约束在一个动态上界和一个恒定下界之间,通过这个动态边界避免出现不稳定和极端的学习率现象,但是需要手动设置其界限,如果设置不当可能会有反效果;Zhuang 等[16]提出了 Adabelief 算法,根据当前梯度方向上的“信念”调整学习率,即通过比较预测梯度和观测梯度之间的差异,实现学习率的自适应缩放,解决了 Adam 算法在训练复杂网络时泛化性与稳定性较差的问题。纵观以上算法,虽然都对 Adam 算法做了改进,但却都基于算法的二阶矩,而 Adam 算法的一阶矩,是一个经典的动量项,是可以被修改的,由此可以使训练效率进一步得到提升。

因此,针对优化算法在训练模型时效率较低的问题,提出了具有快速收敛速度的 Nadabelief 算法。首先,将算法的一阶矩改进为经 Nesterov 加速的动量项,这样在梯度更新时不仅考虑当前的梯度方向,还借助于之前的历史累积梯度来修正更新幅度;接下来,在理论方面证明了其收敛性;最后通过仿真实验验证了算法的性能。

2　算法分析与改进

2.1　自适应算法

在深度学习中,自适应算法是常用的优化算法之一,这类算法可以自动调整学习率,使得模型训练速度更快,稳定性更好。Reddi 等[8]给出了自适应算法的基本框架,见表 1。

表 1　自适应算法框架

Table 1　Framework of adaptive algorithm

算法 1:自适应算法框架
输入:$x_0 \in F$;$\{\alpha_t\}_{t=1}^{T}$;$\{\phi_t,\varphi_t\}_{t=1}^{T}$
初始化:$m_0=0, V_0=0$
对于$t=1,\cdots,T$:
计算梯度:$g_t=\nabla f_t(x_t)$
计算一阶矩:$m_t=\phi_t(g_1,\cdots,g_t)$
计算二阶矩:$V_t=\varphi_t(g_1,\cdots,g_t)$
更新参数:$x_{t+1}=\Pi_{F,\sqrt{V_t}}(x_t-\alpha_t m_t/\sqrt{V_t})$
算法结束

不同自适应算法的一阶矩和二阶矩是不同的,例如在 Adam 算法中,一阶矩和二阶矩分别定义如下:

$$\begin{aligned} m_t &\leftarrow \beta_1 m_{t-1}+(1-\beta_1)g_t \\ v_t &\leftarrow \beta_2 v_{t-1}+(1-\beta_2)g_t^2 \end{aligned} \tag{1}$$

在 Adam 算法的众多变体之中,Adabelief 算法是目前最常用的优化算法之一,见表 2。它依照当前梯度方向上的“信念”调整学习率,将噪声梯度的指数移动平均看作下一时刻梯度的预测。如果观测到的梯度与预测偏差很大,则不“信任”当前的观测值,在下一次更新时减小学习率;如果观测到的梯度接近预测,则“信任”它,并增大学习率。因此,Adabelief 算法的学习率得到

了有效控制,改进了 Adam 算法的泛化性问题。Adabelief 算法的二阶矩为

$$s_t \leftarrow \beta_2 s_{t-1} + (1-\beta_2)(g_t - m_t)^2 + \varepsilon \tag{2}$$

相比于 Adam 算法,Adabelief 算法改进之处在于它修改了 Adam 算法的二阶矩。在 Adam 算法中,更新方向是 $m_t/\sqrt{v_t}$,其中 v_t 代表 g_t^2 的指数移动平均值,g_t 代表梯度,$t \in \{1,2,\cdots,T\}$。而 Adabelief 算法的更新方向是 $m_t/\sqrt{s_t}$,其中 s_t 是 $(g_t-m_t)^2$ 的指数移动平均值。尽管 Adabelief 算法的收敛性能良好,但它只是简单修改了二阶矩,其一阶矩仍然有改进空间,在不损失其泛化能力的前提下,它的收敛速度可以进一步加快,收敛精度也可以进一步提高。

表 2 Adabelief 算法

Table 2 Adabelief algorithm

算法 2:Adabelief 算法
对于 $t=1,\cdots,T$:
计算梯度:$g_t \leftarrow \nabla_x f_t(x_t)$
计算一阶矩:$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
计算二阶矩:$s_t \leftarrow \beta_2 s_{t-1} + (1-\beta_2)(g_t-m_t)^2 + \varepsilon$
修正一阶偏差估计:$\hat{m}_t \leftarrow m_t/(1-\beta_1^t)$
修正二阶偏差估计:$\hat{s}_t \leftarrow s_t/(1-\beta_2^t)$
更新参数:$x_t \leftarrow \Pi_{F,\sqrt{\hat{s}_t}}(x_{t-1} - \alpha \cdot \hat{m}_t/(\sqrt{\hat{s}_t}+\varepsilon))$
算法结束

2.2 基于动量的加速算法

动量梯度下降法是对梯度下降法的一种改进。在训练深度学习模型时,普通的梯度下降法收敛速度相对较慢,且在接近最优值时容易陷入局部最优。而动量法的更新不仅依赖于当前的梯度下降方向,还依赖于之前的更新方向。当算法接近局部最优时,动量法有助于跳出局部最优,从而使得算法更快收敛。具体而言,在迭代时,经典动量更新过程可表示为

$$\begin{aligned} g_t &\leftarrow \nabla_{x_{t-1}} f(x_{t-1}) \\ m_t &\leftarrow \mu m_{t-1} + g_t \\ x_t &\leftarrow x_{t-1} - \eta m_t \end{aligned} \tag{3}$$

Nesterov Accelerated Gradient (NAG) 是经过 Nesterov 加速后的动量梯度下降[17],比梯度下降法的收敛速度更快,其更新过程可以表示为

$$\begin{aligned} g_t &\leftarrow \nabla_{x_{t-1}} f(x_{t-1} - \eta\mu m_{t-1}) \\ m_t &\leftarrow \mu m_{t-1} + g_t \\ x_t &\leftarrow x_{t-1} - \eta m_t \end{aligned} \tag{4}$$

相较于经典动量更新,NAG 的改进在于以“向前看”看到的梯度而不是当前位置梯度去更新,即多了一个本次梯度相对上次梯度的变化量,从而更准确地确定更新方向,这样可以更好地逼近全局最优,也使得 NAG 算法拥有比经典动量更快的收敛速度。为了增强论文的可读性,NAG 算法可以表示为

$$\begin{aligned} g_t &\leftarrow \nabla_{x_{t-1}} f(x_{t-1}) \\ m_t &\leftarrow \mu_t m_{t-1} + g_t \\ \bar{m}_t &\leftarrow g_t + \mu_{t+1} m_t \\ x_t &\leftarrow x_{t-1} - \eta \bar{m}_t \end{aligned} \tag{5}$$

2.3 改进的算法框架

本节主要介绍改进后的 Nadabelief 算法,该算法旨在保留其良好泛化能力的前提下提高其收敛速度。Adabelief 主要由两部分构成,一部分是动量项,另一部分是自适应调整学习率项。经典动量项体现在一阶矩中,而调整学习率项则体现在二阶矩中。Adabelief 虽然调整了二阶矩项,但是一阶矩仍然是一个经典的动量项,而 NAG 机制可以很好地与其结合并加速算法收敛。因此,本文在经典动量项的基础上加入了 NAG 机制,使得更新幅度得到修正,算法的收敛速度进一步加快。

忽略初始化偏差修正项的影响,可以将 Adabelief 的更新规则写为

$$x_{t+1} = x_t - \frac{\alpha_t}{\sqrt{s_t}}(\beta_t m_{t-1} + (1-\beta_t) g_t) \tag{6}$$

接下来用 Nesterov 动量对这个更新过程进行改进,改进后的表达式为

$$\begin{aligned} \bar{m}_t &\leftarrow \beta_{t+1} m_t + (1-\beta_t) g_t \\ x_{t+1} &= x_t - \alpha_t \bar{m}_t/\sqrt{s_t} \end{aligned} \tag{7}$$

考虑算法的初始化偏差修正项,得到改进的算法(Nadabelief)如表 3 所示。

表 3 NAG 改进之后的算法

Table 3 Algorithm improved by NAG

算法 3:Nadabelief 算法
输入:x_0
初始化参数:$m_0 \leftarrow 0; s_0 \leftarrow 0; t \leftarrow 0$
对于 $t=1,\cdots,T$:
计算梯度:$g_t \leftarrow \nabla_x f_t(x_t)$
计算一阶矩:$m_t \leftarrow \beta_t m_{t-1} + (1-\beta_t) g_t$
计算二阶矩:$s_t \leftarrow \beta_2 s_{t-1} + (1-\beta_2)(g_t-m_t)^2 + \varepsilon$
修正一阶偏差估计:$\hat{m}_t \leftarrow m_t/(1-\beta_1^t)$
修正二阶偏差估计:$\hat{s}_t \leftarrow s_t/(1-\beta_2^t)$
加入 NAG 机制:$\bar{m}_t \leftarrow \beta_{t+1}\hat{m}_t + (1-\beta_t) g_t$
更新参数:$x_t \leftarrow \Pi_{F,\sqrt{\hat{s}_t}}(x_{t-1} - \alpha \cdot \bar{m}_t/(\sqrt{\hat{s}_t}+\varepsilon))$
算法结束

3　凸优化的遗憾分析

3.1　最优化问题

深度学习中的模型训练过程本质上就是求解一个最优化问题,即

$$\underset{\theta\in\Theta}{\text{minimize}} f(\theta)$$

其中,θ 为网络参数;Θ 为网络参数的定义域,是一个凸集;f 为需要优化的代价函数,训练目的是将其最小化。

给定一个任意未知的凸代价函数序列 $f_1(\theta)$,$f_2(\theta),\cdots,f_T(\theta)$,目标是预测每个时间 t 的参数 θ_t,并使用未知的成本函数 f_t 来评估这个参数。选择遗憾(Regret)作为算法收敛的衡量指标,即指算法在某个时刻 t 的决策与在所有时刻上的最优决策之间的差距,其定义为

$$R(T)=\sum_{t=1}^{T}f_t(\theta_t)-\min_{\theta\in\Theta}\sum_{t=1}^{T}f_t(\theta)$$

当时间 T 趋于无穷大时,$\lim(R(T)/T)=0$,即当算法对应的 $R(T)$ 有上界时,则认为算法收敛。

3.2　收敛性分析

定理 1　令 $\{x_t\}$ 和 $\{s_t\}$ 为 Nadabelief 算法产生的序列,对于任意的 $t\in[T]$,令 $\alpha_t=\alpha/\sqrt{t}$,$s_t\leqslant s_{t+1}$。假定对于任意的 $x,y\in F$,具有有界直径 $\|x-y\|_\infty\leqslant D_\infty$,若 $f(\theta)$ 为凸函数,且 $\|g_t\|\leqslant G_\infty/2$,则有下列遗憾界:

$$R(T)\leqslant\frac{D_\infty^2\sqrt{T}}{2\alpha(1-\beta_1)}\sum_{i=1}^{d}S_{T,i}^{1/2}+\frac{\alpha\beta_1\sqrt{1+\log T}}{\sqrt{c}(1-\beta_1)^3}\sum_{i=1}^{d}\|g_{1:T,i}^2\|_2+\frac{\alpha D_\infty^2\sqrt{1+\log t}}{2\sqrt{c}}+\frac{D_\infty^2}{2(1-\beta_1)}\sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\beta_{t+1}S_{t,i}^{1/2}}{\alpha_t}\tag{1}$$

证明　由投影运算 $\prod_{F,\sqrt{S_t}}$ 的定义,有

$$x_{t+1}=\prod_{F,\sqrt{S_t}}(x_t-\alpha_t s_t^{-1/2}\bar{m}_t)=\min_{\theta\in F}\|S_t^{1/4}[x-(x_t-\alpha_t s_t^{-1/2}\bar{m}_t)]\|\tag{2}$$

根据 Mamahan 等[18]的理论,令 $y_1=x_{t+1}$,$y_2=x^*$,则有

$$\|S_t^{1/4}(x_{t+1}-x^*)\|^2\leqslant\|S_t^{1/4}(x_t-\alpha_t s_t^{-1/2}\bar{m}_t-x^*)\|=\|S_t^{1/4}(x_t-x^*)\|^2+\|S_t^{-1/4}\alpha_t\bar{m}_t\|^2-2\alpha_t\langle\bar{m}_t,x_t-x^*\rangle=\|S_t^{1/4}(x_t-x^*)\|^2+\|S_t^{-1/4}\alpha_t\bar{m}_t\|^2-2\alpha_t\langle\beta_{t+1}m_t+(1-\beta_t)g_t,x_t-x^*\rangle\tag{3}$$

将式(3)重新组合并利用柯西不等式,得到式(4):

$$\langle g_t,x_t-x^*\rangle\leqslant\frac{1}{2\alpha_t(1-\beta_t)}[\|S_t^{1/4}(x_t-x^*)\|^2-\|S_t^{1/4}(x_{t+1}-x^*)\|^2]+\frac{\alpha_t}{2(1-\beta_t)}\|S_t^{-1/4}\bar{m}_t\|^2-\frac{\beta_{t+1}}{1-\beta_t}\langle m_t,x_t-x^*\rangle\leqslant\frac{1}{2\alpha_t(1-\beta_t)}[\|S_t^{1/4}(x_t-x^*)\|^2-\|S_t^{1/4}(x_{t+1}-x^*)\|^2]+\frac{\alpha_t}{2(1-\beta_t)}\|S_t^{-1/4}\bar{m}_t\|^2+\frac{\alpha_t\beta_{t+1}}{2(1-\beta_t)}\|S_t^{-1/4}m_t\|^2+\frac{\beta_{t+1}}{2\alpha_t(1-\beta_t)}\|S_t^{1/4}(x_t-x^*)\|^2\tag{4}$$

根据遗憾的定义,可得:

$$\sum_{t=1}^{T}f_t(x_t)-f_t(x^*)\leqslant\sum_{t=1}^{T}\langle g_t,x_t-x^*\rangle\leqslant\sum_{t=1}^{T}\{\frac{1}{2\alpha_t(1-\beta_t)}[\|S_t^{1/4}(x_t-x^*)\|^2-\|S_t^{1/4}(x_{t+1}-x^*)\|^2]+\frac{\alpha_t}{2(1-\beta_t)}\|S_t^{-1/4}\bar{m}_t\|^2+\frac{\alpha_t\beta_{t+1}}{2(1-\beta_t)}\|S_t^{-1/4}m_t\|^2+\frac{\beta_{t+1}}{2\alpha_t(1-\beta_t)}\|S_t^{1/4}(x_t-x^*)\|^2\}=\underbrace{\sum_{t=1}^{T}\frac{1}{2\alpha_t(1-\beta_t)}[\|S_t^{1/4}(x_t-x^*)\|^2-\|S_t^{1/4}(x_{t+1}-x^*)\|^2]}_{E_1}+\underbrace{\sum_{t=1}^{T}\frac{\alpha_t}{2(1-\beta_t)}\|S_t^{-1/4}\bar{m}_t\|^2+\sum_{t=1}^{T}\frac{\alpha_t\beta_{t+1}}{2(1-\beta_t)}\|S_t^{-1/4}m_t\|^2}_{E_2}+\underbrace{\sum_{t=1}^{T}\frac{\beta_{t+1}}{2\alpha_t(1-\beta_t)}\|S_t^{1/4}(x_t-x^*)\|^2}_{E_3}\tag{5}$$

E_1 部分、E_2 部分和 E_3 部分分别计算如下:

$$E_1=\sum_{t=1}^{T}\frac{1}{2\alpha_t(1-\beta_t)}[\|S_t^{1/4}(x_t-x^*)\|^2-\|S_t^{1/4}(x_{t+1}-x^*)\|^2]=\frac{1}{2\alpha_1(1-\beta_1)}\|S_1^{1/4}(x_1-x^*)\|^2+\sum_{t=2}^{T}\frac{1}{2\alpha_t(1-\beta_t)}\|S_t^{1/4}(x_t-x^*)\|^2-\sum_{t=2}^{T}\frac{1}{2\alpha_{t-1}(1-\beta_{t-1})}\|S_{t-1}^{1/4}(x_t-x^*)\|^2-\frac{1}{2\alpha_T(1-\beta_T)}\|S_T^{1/4}(x_{T+1}-x^*)\|^2\leqslant\frac{1}{2\alpha_1(1-\beta_1)}\|S_1^{1/4}(x_1-x^*)\|^2+\sum_{t=2}^{T}\frac{1}{2(1-\beta_t)}\|x_t-x^*\|^2\left(\frac{S_t^{1/2}}{\alpha_t}+\frac{S_{t-1}^{1/2}}{\alpha_{t-1}}\right)=\frac{1}{2\alpha_1(1-\beta_1)}\sum_{i=1}^{d}S_{1,i}^{1/2}\|x_{1,i}-x_i^*\|^2+$$

$$\sum_{t=2}^{T}\sum_{i=1}^{d}\| x_{t,i}-x_i^*\|^2\left(\frac{S_{t,i}^{1/2}}{\alpha_t}+\frac{S_{t-1,i}^{1/2}}{\alpha_{t-1}}\right)\leqslant \frac{D_\infty^2\sqrt{T}}{2\alpha(1-\beta_1)}\sum_{i=1}^{d}S_{T,i}^{1/2} \tag{6}$$

$$\begin{aligned}E_2=&\sum_{t=1}^{T}\frac{\alpha_t}{2(1-\beta_t)}\|S_t^{-1/4}\bar{m}_t\|^2+\\&\sum_{t=1}^{T}\frac{\alpha_t\beta_{t+1}}{2(1-\beta_t)}\|S_t^{-1/4}m_t\|^2=\\&\sum_{t=1}^{T}\frac{\alpha_t}{2(1-\beta_t)}\|S_t^{-1/4}(\beta_{t+1}m_t+(1-\beta_t)g_t)\|^2+\\&\sum_{t=1}^{T}\frac{\alpha_t\beta_{t+1}}{2(1-\beta_t)}\|S_t^{-1/4}m_t\|^2\leqslant\\&\sum_{t=1}^{T}\frac{\alpha_t\beta_{t+1}^2}{2(1-\beta_t)}\|S_t^{-1/4}m_t\|^2+\sum_{t=1}^{T}\frac{\alpha_t\beta_{t+1}}{2}\|S_t^{-1/4}m_t\|^2+\\&\sum_{t=1}^{T}\frac{\alpha_t\beta_{t+1}}{2(1-\beta_t)}\|S_t^{-1/4}m_t\|^2+\sum_{t=1}^{T}\frac{\alpha_t(1-\beta_t)}{2}\|S_t^{-1/4}g_t\|^2+\\&\sum_{t=1}^{T}\frac{\alpha_t\beta_{t+1}}{2}\|S_t^{-1/4}g_t\|^2\leqslant\frac{\alpha\beta_1\sqrt{1+\log T}}{\sqrt{c}(1-\beta_1)^3}\sum_{i=1}^{d}\|g_{1:T,i}^2\|_2+\\&\frac{\alpha D_\infty^2\sqrt{1+\log t}}{2\sqrt{c}}\end{aligned} \tag{7}$$

$$E_3=\sum_{t=1}^{T}\frac{\beta_{t+1}}{2\alpha_t(1-\beta_t)}\|S_t^{1/4}(x_t-x^*)\|^2\leqslant \frac{D_\infty^2}{2(1-\beta_1)}\sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\beta_{t+1}S_{t,i}^{1/2}}{\alpha_t} \tag{8}$$

因此,遗憾界为

$$\begin{aligned}R(T)=&\sum_{t=1}^{T}f_t(x_t)-f_t(x^*)\leqslant\sum_{t=1}^{T}\langle g_t,x_t-x^*\rangle\leqslant\\&\sum_{t=1}^{T}\frac{1}{2\alpha_t(1-\beta_t)}\left[\|S_t^{1/4}(x_t-x^*)\|^2-\|S_t^{1/4}(x_{t+1}-x^*)\|^2\right]+\\&\sum_{t=1}^{T}\frac{\alpha_t}{2(1-\beta_t)}\|S_t^{-1/4}\bar{m}_t\|+\sum_{t=1}^{T}\frac{\alpha_t\beta_{t+1}}{2(1-\beta_t)}\|S_t^{-1/4}m_t\|^2+\\&\sum_{t=1}^{T}\frac{\beta_{t+1}}{2\alpha_t(1-\beta_t)}\|S_t^{1/4}(x_t-x^*)\|^2\leqslant\\&\frac{D_\infty^2\sqrt{T}}{2\alpha(1-\beta_1)}\sum_{i=1}^{d}S_{T,i}^{1/2}+\frac{\alpha\beta_1\sqrt{1+\log T}}{\sqrt{c}(1-\beta_1)^3}\sum_{i=1}^{d}\|g_{1:T,i}^2\|_2+\\&\frac{\alpha D_\infty^2\sqrt{1+\log t}}{2\sqrt{c}}+\frac{D_\infty^2}{2(1-\beta_1)}\sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\beta_{t+1}S_{t,i}^{1/2}}{\alpha_t}\end{aligned} \tag{9}$$

整理得:

$$R(T)\leqslant\frac{D_\infty^2\sqrt{T}}{2\alpha(1-\beta_1)}\sum_{i=1}^{d}S_{T,i}^{1/2}+\frac{\alpha\beta_1\sqrt{1+\log T}}{\sqrt{c}(1-\beta_1)^3}\sum_{i=1}^{d}\|g_{1:T,i}^2\|_2+\frac{\alpha D_\infty^2\sqrt{1+\log t}}{2\sqrt{c}}+\frac{D_\infty^2}{2(1-\beta_1)}\sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\beta_{t+1}S_{t,i}^{1/2}}{\alpha_t}$$

至此,证明了 $R(T)$ 存在上界,Nadabelief 算法收敛。

4 仿真实验与结果分析

为了测试 Nadabelief 算法的性能,本文进行了两组实验,一组是 Logistic 回归的凸实验,另一组是 CNN 图像分类与 LSTM 语言建模的非凸实验(深度学习训练往往都是在此环境下进行的)。在所有实验中,都将提出的 Nadabelief 算法与 Adam 算法、Nadam 算法和 Adabelief 算法的性能进行对比。

4.1 Logistic 回归

利用 Logistic 回归,在 MNIST 数据集上比较算法在凸问题上的性能。MNIST 数据集包含 10 个类共 60 000 张图片,这些图片由 250 个人手写不同的数字所组成。在实验中,β_1 和 β_2 分别被设置为 0.9 和 0.999。由图 1 可知,Nadabelief 算法在 logistic 回归中的收敛速度优于其他算法,算法的损失函数最小。

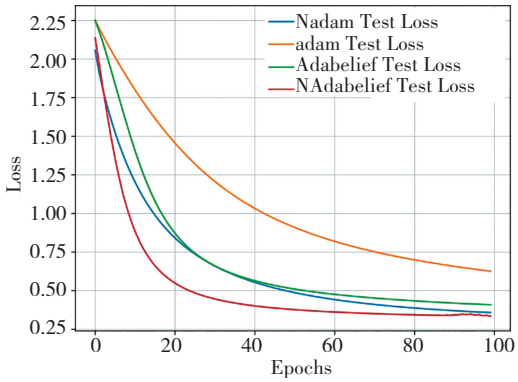


图 1 Logistic 回归实验

Fig. 1 Logistic regression experiment

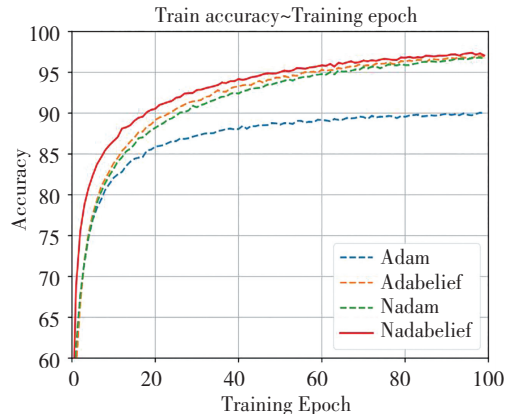
4.2 图像分类

为了测试 Nadabelief 算法在图像分类领域的有效性,使用 Cifar10 与 Cifar100 这两个基本数据集对其进行实验,在实验过程中使用的 CNN 模型为 Resnet20。

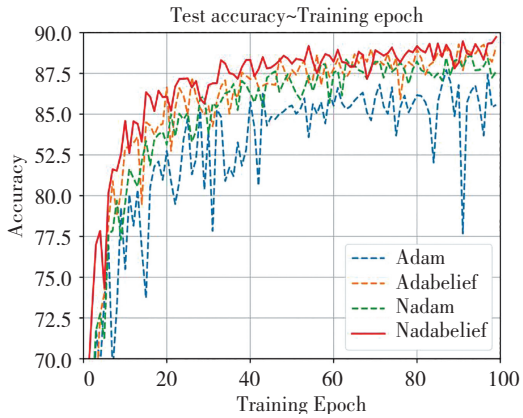
Cifar10 数据集是一个用于识别物体的数据集,共包含有 60 000 张大小为 32 * 32 的彩色 RGB 图片,其中 50 000 张用于训练,10 000 张用于测试,数据集共分为 10 个类别。在实验中,Adam、Nadam、Adabelief 和 Nadabelief 算法的 β_1 均取为 0.9,β_2 均取为 0.999,学习率为 0.001,ε 值为 e^{-8}。

实验结果如图 2 所示,Nadabelief 算法在 Cifar10 数据集上的收敛速度和测试精度都优于与之比较的其他算法。此外,Adam 算法的振荡最剧烈,在所有对比算法中性能最差。

算法的损失函数见图 3，Nadabelief 算法的损失值下降最快。



（a）训练精确度



（b）测试精确度

图 2　Cifar10 数据集上各算法在 CNN 模型中的训练精确度和测试精确度

Fig. 2　The training accuracy and testing accuracy of each algorithm on the Cifar10 dataset in the CNN model

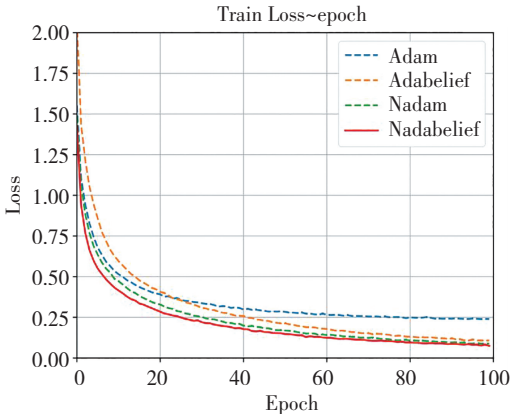
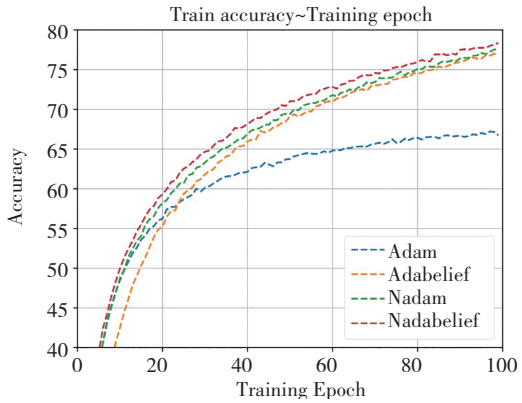


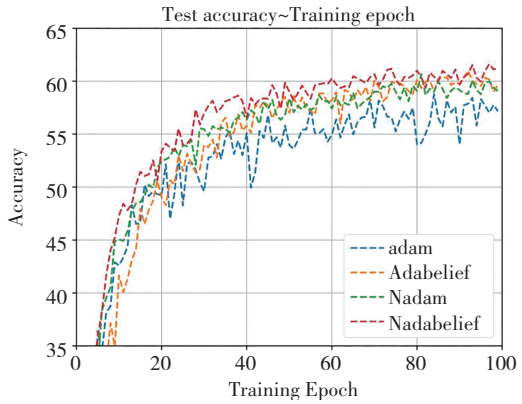
图 3　Cifar10 数据集上各算法的训练损失

Fig. 3　Training losses of different algorithms on the Cifar10 dataset

与 Cifar10 数据集类似，Cifar100 数据集同样也是使用非常广泛的图像分类数据集。与 Cifar10 数据集不同的是，Cifar100 数据集有 100 个类别，且这 100 个类被分为 20 个大类。每个类包含 600 个图像，其中 500 张用于训练，100 张用于测试。在实验中，各优化算法的 β_1 均取为 0.9，β_2 均取为 0.999，学习率为 0.001，ε 值为 e^{-8}。实验结果如图 4、图 5 所示，可以看到，Nadabelief 算法在 Cifar100 数据集上的性能也优于其他算法。



（a）训练精确度



（b）测试精确度

图 4　Cifar100 数据集上各算法在 CNN 模型中的训练精度和测试精度

Fig. 4　The training accuracy and testing accuracy of each algorithm on the Cifar100 dataset in the CNN model

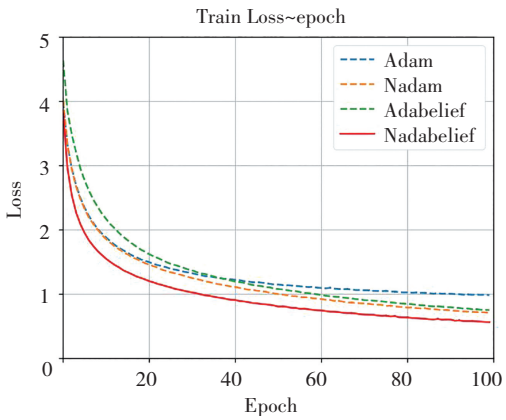


图 5　Cifar100 数据集上各算法的训练损失

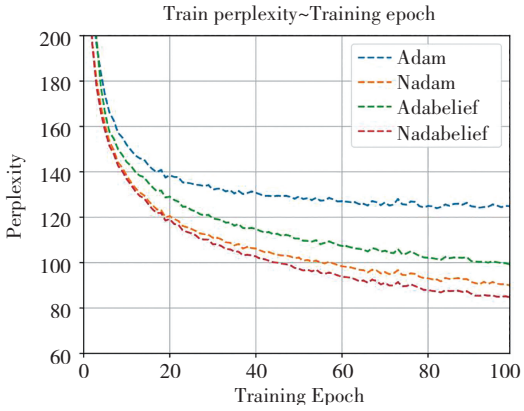
Fig. 5　Training losses of different algorithms on the Cifar100 dataset

4.3 语言建模

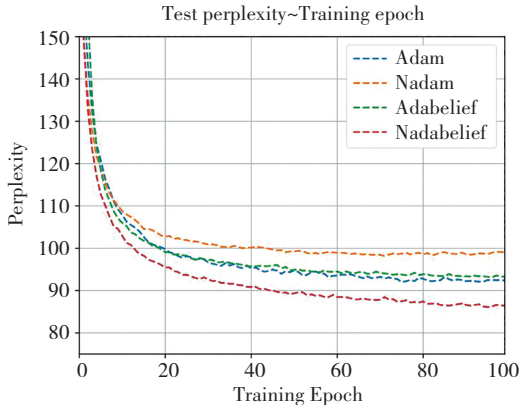
接下来使用长短时记忆网络(LSTM)来进行语言建模实验。语言模型是自然语言处理的基础,其任务就是预测每个句子在语言中出现的概率。本文使用的数据集为 Penn TreeBank(PTB)数据集。实验采用的是 2 层 LSTM 模型。

PTB 数据集是目前自然语言处理中使用非常广泛的数据集。该数据集共分为 24 个部分,各部分功能分割如下:0—18 的部分用于训练,19—21 的部分用于验证,22—24 的部分用于测试。在实验中,算法的 β_1 均取为 0.9,β_2 均取为 0.999,学习率均取为 0.01,ε 值为 e^{-12}。采用困惑度(Perplexity)来衡量语言模型的性能,Perplexity 越小,表示模型的预测效果越好。

从图 6 中可以看出:Nadabelief 算法在训练集和测试集上的 Perplexity 最低,验证了该算法相比其他算法的预测效果更好。



(a) 训练困惑度



(b) 测试困惑度

图 6 PTB 数据集上各算法在 LSTM 模型中的训练困惑度和测试困惑度

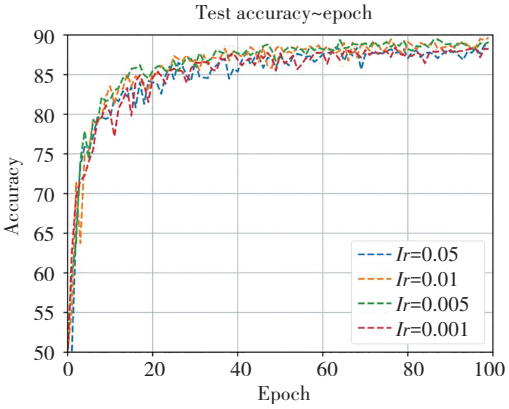
Fig. 6 The training perplexity and testing perplexity of algorithms on the PTB dataset in the LSTM model

4.4 结果分析

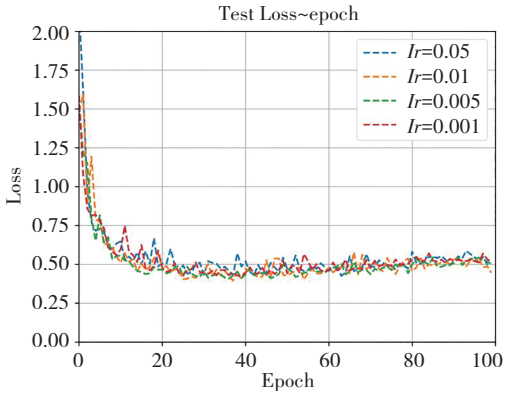
图像分类实验中,当算法使用相同模型进行训练时,Nadabelief 由于 NAG 机制,它的收敛速度和稳定性都有提升。观察图 2 和图 4 的精确度图像可以看出,在相同的迭代次数内,Nadabelief 所达到的精确度比 Adam、Adabelief 算法更高,Adam 算法在后期出现了不稳定的振荡现象,Nadabelief 很好地规避了这一问题。观察图 3 和图 5 的损失值图像,Nadabelief 在相同的迭代次数内取得了更低的损失值,模型的拟合效果更好。语言建模实验中,Nadabelief 算法所取得的困惑度也是最低的,多个实验验证了所提出算法的性能优异。

在算法复杂度方面,虽然 Adam 和 Nadabelief 算法所能达到的下界是相同的,但是 Nadabelief 算法的空间复杂度要略高于 Adam,因为 Adam 需要存储动量和其平方项的累积值,以及参数的一阶和二阶矩估计,而 Nadabelief 算法除了存储上述参数值外,还需要额外存储 Nesterov 动量的估计,这也是在以后的工作中可以进行研究并改进之处。

在 Cifar10 数据集上测试算法在不同的初始学习率下的性能。从图 7 可以清楚地看到,Nadabelief 算法在训练最后阶段的准确率和损失值相差不大。这进一步证明了算法对不同学习率的鲁棒性。



(a) 测试精度



(b) 测试损失

图 7 学习率不同时算法在 Cifar10 数据集上的测试精度和测试损失

Fig. 7 Testing accuracy and loss of algorithm with different learning rates on Cifar10

综上,本算法实现了在模型训练时收敛速度的进一步提升,且稳定性与泛化性良好。

5 结　论

本文以自适应算法的一阶矩为突破点,通过将 Nesterov 加速机制应用于经典动量项,提出了一种具有快速收敛的 Nadabelief 算法,提升了深度学习模型训练时的效率。

使用 Adabelief 算法代替传统的 Adam 算法,提高了优化算法的泛化性与稳定性。采用 Nesterov 加速机制,在保持原有 Adabelief 算法泛化能力的同时使得算法兼具更快的收敛速度和更好的收敛精度。理论分析得到了 Nadabelief 算法的遗憾界;同时,确保了所提出算法的收敛性。为了验证算法的性能,在 Logistic 回归、Cifar10 与 Cifar100 图像分类数据集以及 Penn TreeBank 语言建模数据集上进行实验,通过与 Adam、Nadam、Adabelief 算法的比较,显示所提出的算法效果优于与之比较的算法。验证了 Nadabelief 算法的优越性,因此,使用 Nadabelief 优化算法训练深度学习模型是有效的。

虽然仿真实验清楚地验证了算法在非凸情况下收敛,但是在此情况下的理论证明还有待进一步建立。在未来的工作中,还将尝试使用提出的算法来训练复杂的神经网络,以解决更具挑战性的问题。

参考文献(References):

[1] ALZUBAIDI L, ZHANG J, HUMAIDI A J, et al. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions[J]. Journal of Big Data, 2021, 8(1): 53.

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

[3] NESTEROV Y E. A method for solving the convex programming problem with convergence rate O $\left(\frac{1}{k^2}\right)$ [J]. Dokl Akad Nauk SSSR, 1983, 269: 543-547.

[4] 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 Machine Learning. PMLR, 2013: 1139-1147.

[5] KINGMA D P, BA J. Adam: A method for stochastic optimization[C]//The 3rd International Conference for Learning Representations. SanDiego: ICLR, 2015: 1-15.

[6] DUCHI J, HAZAN E, SINGER Y. Adaptive subgradient methods for online learning and stochastic optimization [J]. Journal of Machine Learning Research, 2011, 12(7): 2121-2159.

[7] TANG H, GAN S, AWAN A A, et al. 1-bit Adam: Communication efficient large-scale training with Adam' s convergence speed[C]//International Conference on Machine Learning. PMLR, 2021: 10118-10129.

[8] REDDI S J, KALE S, KUMAR S. On the convergence of Adam and beyond[C]//International Conference for Learning Representations. ICLR, 2018: 1-23.

[9] 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. New York: ACM, 2017: 4151-4161.

[10] REYAD M, SARHAN A M, ARAFA M. A modified Adam algorithm for deep neural network optimization [J]. Neural Computing and Applications, 2023, 35(23): 17095-17112.

[11] LU J. AdaSmooth: An adaptive learning rate method based on effective ratio [C]//Sentiment Analysis and Deep Learning. Singapore: Springer Nature Singapore, 2023: 273-293.

[12] DUMAN B, ALISÜZEN A. A study on deep learning based classification of flower images [J]. International Journal of Advanced Networking and Applications, 2022, 14(2): 5385-5389.

[13] AGARWAL A K, KIRAN V, JINDAL R K, et al. Optimized transfer learning for dog breed classification[J]. International Journal of Intelligent Systems and Applications in Engineering, 2022, 10(1s): 18-22.

[14] YANG L, CAI D. AdaDB: An adaptive gradient method with data-dependent bound[J]. Neurocomputing, 2021, 419(2): 183-189.

[15] FANG R, LI D, SHEN X. Distributed online adaptive subgradient optimization with dynamic bound of learning rate over time - varying networks [J]. IET Control Theory & Applications, 2022, 16(18): 1834-1846.

[16] ZHUANG J, TANG T, DING Y, et al. AdaBelief optimizer: adapting stepsizes by the belief in observed gradients [C]// Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020: 18795-18806.

[17] DOZAT T. Incorporating nesterov momentum into adam[C]// International Conference on Learning Representations. ICLR, 2016. 1-4.

[18] MCMAHAN H B, STREETER M. Adaptive bound optimization for online convex optimization [C]//Proceedings of the 23rd Annual Conference on Learning Theory. COLT, 2010: 244-256.

责任编辑:李翠薇