Advances in Applied Mathematics 应用数学进展, 2024, 13(7), 3182-3192
Published Online July 2024 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2024.137304

Hans汉斯

一种新型的分数阶梯度下降法在深度神经网络中的应用

吴　昊

南京航空航天大学数学学院，江苏　南京

收稿日期：2024年6月15日；录用日期：2024年7月9日；发布日期：2024年7月17日

摘　要

文章基于Caputo分数阶微积分，提出了一种新型的适用于神经网络模型训练的分数阶梯度下降法。该算法通过改变积分区间下界，成功将分数阶阶次拓展到了(0, 2)区间，增加了阶次的选择范围，同时，本文基于梯度裁剪机制，从遗憾函数的角度证明了该算法的收敛性，保证了算法的理论可行性。最后，基于CIFAR-10公开数据集的数值实验表明，在选择了合适的阶次的情况下，本文所提出的算法相比于传统的整数阶梯度法，能够获得更快的收敛速度和更高的收敛精度。

关键词

Caputo分数阶微积分，梯度下降法，遗憾，深度神经网络

A Novel Fractional-Order Gradient Descent Method with Its Application in Deep Neural Network

Hao Wu

School of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing Jiangsu

Received: Jun. 15th, 2024; accepted: Jul. 9th, 2024; published: Jul. 17th, 2024

Abstract

This study introduces a novel fractional gradient descent algorithm based on Caputo fractional calculus which is tailored for training neural network models. By adjusting the lower limit of the integral interval, the proposed algorithm extends the fractional order to the (0, 2) range, thereby

文章引用: 吴昊. 一种新型的分数阶梯度下降法在深度神经网络中的应用[J]. 应用数学进展, 2024, 13(7): 3182-3192.
DOI: 10.12677/aam.2024.137304

enhancing the choices of fractional order. Concurrently, this work proves the convergence of the proposed algorithm in detail from the perspective of the regret function based on the gradient clipping mechanism, affirming its theoretical validity. Finally, the numerical experiment based on the publicly available CIFAR-10 dataset, reveals that the proposed algorithm outperforms conventional integer-order gradient method in terms of both convergence speed and convergence accuracy when operated at an optimal order.

Keywords

Caputo Fractional Calculus, Gradient Descent Method, Regret, Deep Neural Network

Copyright © 2024 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 引言

神经网络是一种人工智能技术，其目的是模仿人脑神经元的工作原理，实现对各种信息的处理和识别。虽然神经网络的发展经历了一段时间的沉寂，但近年来，神经网络模型得到了越来越多的关注，重新回归到了研究者的视野当中。卷积神经网络(CNN) [1]和循环神经网络(RNN) [2]是深度神经网络的两大基础架构，前者是一类专门处理图像数据的网络模型，它使用卷积操作代替前馈网络中的乘法操作，逐层提取图像数据的特征，模仿了人脑的视觉区域处理图像的过程，从而在图像识别，目标检测等方面实现了更好的效果，而后者的结构近似若干个前馈网络的串联，通过隐藏层的连接达到信息流动的效果，独特的结构使得后面的时间步依然可以捕获到前面时间步的信息，因此非常适合处理时序数据，已成功应用于语音识别、语音合成、机器翻译等领域。目前，各种新型网络模型层出不穷，其中，最具代表性的便是分别从 RNN 和 CNN 的基础发展而来的 Transformer [3]和 ResNet [4]。Transformer 进一步探索了人脑的结构，引入了注意力机制，提高了信息处理效率，在文本生成和机器翻译等领域取得了巨大的成功，ResNet 在输入层和输出层之间添加了快捷连接(Shortcut Connection)，极大地缓解了深层网络难以训练的问题。

梯度下降法是训练神经网络模型的常见算法。以监督学习为例，首先研究人员需要收集大量的训练数据，同时为训练数据添加标签，然后根据神经网络接收输入数据后得到的输出值与真实值构造损失函数，最后通过计算损失函数的梯度指导神经网络每一步的训练。然而，仿真实验表明，使用梯度下降法训练神经网络不仅收敛速度慢，而且收敛精度较低，更重要的是，面对复杂的损失函数曲面，普通的梯度下降法易于陷入局部极小值难以逃脱，极大地影响了神经网络的性能。

分数阶微分的概念诞生于 1695 年洛必达和莱布尼茨关于非整数阶次微分的思考，距今已有 300 多年。如今，分数阶微积分已成功应用于自动控制[5]、时序预测[6] [7]、模式识别[8] [9]等领域，表现出了比整数阶更加优异的性能。分数阶微分是整数阶微分的自然推广，因此在梯度下降法中引入阶次的概念不失为一种可行的操作，与整数阶微分相比，分数阶微分具有非局部特性，使用分数阶梯度有利于整合更多的信息指导参数寻优。四川大学蒲亦非教授[10]首先对分数阶梯度下降法展开了研究，为了构造分数阶梯度下降法格式，蒲亦非团队很自然地用黎曼刘维尔分数阶微分替换整数阶梯度，但是，数值实验表明该梯度下降法具有明显的缺陷，即使对于具有唯一全局最优点的凸函数来说，算法依旧会收敛到多个值，无法判断真实的最小值点，原因在于分数阶微分是在整个区间上进行运算，具有长记忆特性，因此分数

阶导数为 0 的点与阶次和初始值有关，并不唯一。同理，基于卡普托分数阶微分定义的分数阶梯度下降法也具有相同的缺点。为了解决这一问题，文献[11]在研究过程中设计了新型分数阶梯度下降法，通过缩小分数阶微分的运算区间，消除了其长记忆特性，保证了算法在收敛的情况下只会收敛到唯一极小值点，在此基础上，又对算法进行了高阶截断的改进，降低了目标函数的选取要求。针对相同的问题，文献[12]同样投入了大量的研究工作，试图构造一种能够收敛到真实全局最优的分数阶梯度下降法，论文中提出了三种可能方案：缩小分数阶微分运算区间，高阶截断以及设计变阶次分数阶梯度下降法。第一种方法相当于文献[11]的拓展，使得运算区间的下限替换为前 k 步的迭代结果，提高了算法的灵活性，拓宽了算法的改进空间，然而其证明部分存在错误，无法得出该分数阶梯度法在收敛时必定收敛到真实极值的结论，本文将从另一角度给出正确的收敛性证明。第二种方法简化了算法形式，有效加速了算法的收敛过程。第三种方法使微分阶次在迭代过程中不断改变，算法在迭代后期趋近于整数阶算法，不仅解决了无法收敛到真实极值点的问题，而且提高了迭代过程的稳定性，数值实验表明其具有潜在的应用价值。文献[13]提出了一种分数阶全局学习机，它将神经网络中参数寻优的过程分为了下降和上升两个阶段，在下降阶段使用当前位置的整数阶梯度指导参数下降，同时在上升阶段利用分数阶阶梯度逃离局部极值以达到全局寻优的效果，文献通过大量的数值实验验证了分数阶学习机的有效性。上述方法中算法阶次仍限制在(0, 1)区间上，本文将给出一种新型的区间下界选择方法，以扩展阶次的选择范围，同时给出详细的收敛性证明，并通过数值实验验证算法的有效性。

本文的剩余部分安排如下。第二节介绍 Caputo 分数阶微分的基础知识和分数阶梯度下降法的具体形式，第三节从遗憾函数的角度证明所提算法在凸函数条件下的收敛性，第四节通过基于 CIFAR-10 数据集的数值实验验证了新型分数阶梯度下降法具有更快的收敛速度。第五节为全文的总结。

2. 基础知识

2.1. Caputo 分数阶导数

m 阶可微函数 $f(x)$ 的 Caputo 分数阶导数定义为以下形式：

$$ {}_aD_x^{\alpha} f(x) = \frac{1}{\Gamma(m-\alpha)} \int_a^x \frac{\mathrm{d}^m f(\tau)}{\mathrm{d}\tau^m} \cdot (x-\tau)^{m-\alpha-1} \mathrm{d}\tau, (m-1<\alpha<m) \tag{1} $$

其中，a 表示固定的积分下界，α 为分数阶阶次，m 是正整数，$\Gamma(\cdot)$ 表示 gamma 函数。

进一步，若 $f(x)$ 拥有泰勒展开式，那么将泰勒展开式代入(1)式右端再进行分部积分处理则可以得到：

$$ {}_aD_x^{\alpha} f(x) = \sum_{i=m}^{\infty} \frac{\Gamma(\alpha-m+1)}{\Gamma(i-m+1)\Gamma(\alpha-i+1)} \cdot \frac{\mathrm{d}^i f(x)}{\mathrm{d}x^i} \cdot \frac{(x-a)^{i-\alpha}}{\Gamma(i-\alpha+1)} \tag{2} $$

2.2. 分数阶梯度下降法

普通的梯度下降法可表示成以下形式：

$$ x_{t+1} = x_t - \eta \cdot \nabla f(x_t) \tag{3} $$

其中，η 表示学习率，$\nabla f(x)$ 为 $f(x)$ 在 x 处的梯度。

为了构造分数阶梯度下降法，则需要使用分数阶导数指导参数下降。当 $0<\alpha<1$ 时，(2)式的具体形式为：

$$ {}_aD_x^{\alpha} f(x) = \sum_{i=1}^{\infty} \frac{\mathrm{d}^i f(x)}{\mathrm{d}x^i} \cdot \frac{(x-a)^{i-\alpha}}{\Gamma(i-\alpha+1)} \tag{4} $$

在面对诸如神经网络等较复杂的应用场景时，二阶及以上的导函数计算将十分占用计算资源，严重减慢参数的更新速度，因此只取$i=1$一项，即：

$$ {}_{a}D_{x}^{\alpha} f(x)=\frac{(x-a)^{1-\alpha}}{\Gamma(2-\alpha)}\cdot\frac{\mathrm{d}f(x)}{\mathrm{d}x} \tag{5} $$

此时，分数阶梯度下降法可表示为：

$$ x_{t+1}=x_t-\eta\cdot {}_{a}D_{x}^{\alpha} f(x_t)=x_t-\eta\cdot\frac{(x_t-a)^{1-\alpha}}{\Gamma(2-\alpha)}\cdot\frac{\mathrm{d}f(x)}{\mathrm{d}x_t} \tag{6} $$

然而，该分数阶梯度算法仍有缺陷，蒲亦非团队[10]的实验结果表明固定的积分下界 a 会导致(6)式无法收敛到真实的极值点，因此需要设计随着迭代过程自适应调整的积分下界a_t。考虑到在参数迭代过程中参数值是不断变化的，因此使a_t与历史参数值相关联不失为一种简单可行的方法，例如，可取$a_t=x_{t-1}$。但是，当$x_{t-1}>x_t$时，$(x_t-x_{t-1})^{1-\alpha}$可能不再是一个正整数，这会阻碍参数的迭代，所以，a_t应严格满足$a_t<x_t$。为了满足这一要求，本文做出以下修改，即：

$$ a_t=\begin{cases} x_{t-1}-c, x_{t-1}\le x_t \\ 2x_t-x_{t-1}-c, x_{t-1}>x_t \end{cases} \tag{7} $$

其中，c 为一固定的常数。当$x_{t-1}\le x_t$时，$a_t=x_{t-1}-c$，在$x_{t-1}=x_t$的情况下，这将避免迭代过程陷入停滞，当$x_{t-1}>x_t$时，$a_t=2x_t-x_{t-1}-c$，其中，$2x_t-x_{t-1}$是x_{t-1}关于x_t的对称点，这使得每一次迭代时区间长度在形式上保持一致。将a_t替换(2.6)式中的a，就得到了本文的分数阶梯度下降法：

$$ x_{t+1}=x_t-\eta\cdot {}_{x_{t-1}}D_{x_t}^{\alpha} f(x) \tag{8} $$

其中，

$$ {}_{x_{t-1}}D_{x_t}^{\alpha} f(x)=\frac{(|x_t-x_{t-1}|+c)^{1-\alpha}}{\Gamma(2-\alpha)}\cdot\frac{\mathrm{d}f(x)}{\mathrm{d}x_t} \tag{9} $$

观察(9)的形式可以发现，当$1\le\alpha<2$时，该算法仍旧可以更新参数，且当$\alpha=1$时退化为普通的梯度下降法，因此，该算法成功将阶次选择范围扩大至(0, 2)区间。

3. 主要结果

3.1. 遗憾函数

一般来说，使用梯度下降方法训练神经网络模型的过程可视作一个在线学习过程，在 t 时刻，优化器从解空间Ω中选取一个解θ_t，同时人为地选取一个损失函数$L_t(\cdot)$，之后优化器根据损失函数更新模型并获得下一个时刻的解θ_{t+1}。遗憾函数是评价在线学习模型的常用指标，其数学表达形式为：

$$ R(T)=\sum_{t=1}^{T}\left[L_t(\theta_t)-L_t(\theta^*)\right] \tag{10} $$

其中，$\theta^*=\arg\min_{\theta\in\Omega}\sum_{t=1}^{T}L_t(\theta)$。遗憾函数衡量了 T 时刻后模型累积的损失与理想模型损失之间的差值，差值越小，则训练模型越接近理想模型，在线学习的目的是最小化积累的损失，最小化损失就等价于最小化遗憾。在线学习算法通常希望达到次线性的遗憾，即：

$$ \lim_{T\to\infty}\frac{R(T)}{T}\to 0 \tag{11} $$

具备次线性遗憾的算法也被称为满足 Hannan 一致性[14]。

3.2. 收敛性定理及证明

定理 1 给定$\{\theta_t\}$为使用分数阶梯度下降法训练神经网络得到的每一步的迭代结果，那么分数阶梯度算法具有次线性遗憾如果其满足如下假设：

1) 损失函数为凸函数；

2) $\left\| {}_{\theta_{t-1}}D_{\theta_t}^{\alpha}L(\theta_t)\right\| \le G$，即任一参数的分数阶梯度有界；

3) 对于解空间Ω中任意的两个参数θ_m,θ_n，有$\|\theta_m-\theta_n\| \le D$；

4) $\left\| {}_{\theta_{t-1,i}}D_{\theta_{t,i}}^{\alpha}L(\theta_t)\right\| \le 1$，即限制分数阶梯度的大小。

证明过程如下：

在凸函数假设下，有：

$$\begin{aligned} L_t(\theta_t)-L_t(\theta^*) &\le \sum_{i=1}^{d}\frac{\mathrm{d}L_t(\theta_t)}{\mathrm{d}\theta_{t,i}}\cdot\left(\theta_{t,i}-\theta_i^*\right) \\ &\le \sum_{i=1}^{d}\Gamma(2-\alpha)\cdot\left(\left|\theta_{t,i}-\theta_{t-1,i}\right|+c\right)^{\alpha-1}{}_{\theta_{t-1,i}}D_{\theta_{t,i}}^{\alpha}L_t(\theta_t)\cdot\left(\theta_{t,i}-\theta_i^*\right) \end{aligned} \tag{12}$$

根据更新公式$\theta_{t+1,i}=\theta_{t,i}-\eta_t\cdot{}_{\theta_{t-1,i}}D_{\theta_{t,i}}^{\alpha}L(\theta_t)$，其中，$\eta_t=\frac{\eta}{\sqrt{t}}$，可以得到：

$${}_{\theta_{t-1,i}}D_{\theta_{t,i}}^{\alpha}L(\theta_t)\cdot\left(\theta_{t,i}-\theta_i^*\right)=\frac{1}{2\eta_t}\left[\left(\theta_{t,i}-\theta_i^*\right)^2-\left(\theta_{t+1,i}-\theta_i^*\right)^2\right]+\frac{\eta_t}{2}\left[{}_{\theta_{t-1,i}}D_{\theta_{t,i}}^{\alpha}L(\theta_t)\right]^2 \tag{13}$$

因此，

$$\begin{aligned} R(T) &= \sum_{t=1}^{T}\left[L(\theta_t)-L(\theta^*)\right] \\ &\le \sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\Gamma(2-\alpha)}{2\eta_t}\left(\left|\theta_{t,i}-\theta_{t-1,i}\right|+c\right)^{\alpha-1}\left[\left(\theta_{t,i}-\theta_i^*\right)^2-\left(\theta_{t+1,i}-\theta_i^*\right)^2\right] \\ &\quad+\sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\eta_t\Gamma(2-\alpha)}{2}\left(\left|\theta_{t,i}-\theta_{t-1,i}\right|+c\right)^{\alpha-1}\left[{}_{\theta_{t-1,i}}D_{\theta_{t,i}}^{\alpha}L(\theta_t)\right]^2 \end{aligned} \tag{14}$$

首先处理第一部分，

$$\begin{aligned} \Delta_1 &= \sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\Gamma(2-\alpha)}{2\eta_t}\left(\left|\theta_{t,i}-\theta_{t-1,i}\right|+c\right)^{\alpha-1}\left[\left(\theta_{t,i}-\theta_i^*\right)^2-\left(\theta_{t+1,i}-\theta_i^*\right)^2\right] \\ &= \sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\Gamma(2-\alpha)}{2\eta_t}\left(\left|\theta_{t,i}-\theta_{t-1,i}\right|+c\right)^{\alpha-1}\left(\theta_{t,i}-\theta_i^*\right)^2 \\ &\quad-\sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\Gamma(2-\alpha)}{2\eta_t}\left(\left|\theta_{t,i}-\theta_{t-1,i}\right|+c\right)^{\alpha-1}\left(\theta_{t+1,i}-\theta_i^*\right)^2 \\ &= \sum_{i=1}^{d}\frac{\Gamma(2-\alpha)}{2\eta}\left(\left|\theta_{1,i}-\theta_{0,i}\right|+c\right)^{\alpha-1}\left(\theta_{1,i}-\theta_i^*\right)^2 \\ &\quad+\sum_{t=2}^{T}\sum_{i=1}^{d}\frac{\Gamma(2-\alpha)}{2\eta_t}\left(\left|\theta_{t,i}-\theta_{t-1,i}\right|+c\right)^{\alpha-1}\left(\theta_{t,i}-\theta_i^*\right)^2 \\ &\quad-\sum_{t=1}^{T-1}\sum_{i=1}^{d}\frac{\Gamma(2-\alpha)}{2\eta_t}\left(\left|\theta_{t,i}-\theta_{t-1,i}\right|+c\right)^{\alpha-1}\left(\theta_{t+1,i}-\theta_i^*\right)^2 \\ &\quad-\sum_{i=1}^{d}\frac{\sqrt{T}\cdot\Gamma(2-\alpha)}{2\eta}\left(\left|\theta_{T,i}-\theta_{T-1,i}\right|+c\right)^{\alpha-1}\left(\theta_{T+1,i}-\theta_i^*\right)^2 \end{aligned} \tag{15}$$

由于最后一项恒为负值，因此可以对 Δ_1 进行放缩处理，即：

$$\begin{aligned}\Delta_1 \le & \sum_{i=1}^{d}\frac{\Gamma(2-\alpha)}{2\eta}\left(\left|\theta_{1,i}-\theta_{0,i}\right|+c\right)^{\alpha-1}\left(\theta_{1,i}-\theta_i^*\right)^2+\sum_{t=2}^{T}\sum_{i=1}^{d}\frac{\Gamma(2-\alpha)}{2\eta_t}\left(\left|\theta_{t,i}-\theta_{t-1,i}\right|+c\right)^{\alpha-1}\left(\theta_{t,i}-\theta_i^*\right)^2 \\ & -\sum_{t=1}^{T-1}\sum_{i=1}^{d}\frac{\Gamma(2-\alpha)}{2\eta_t}\left(\left|\theta_{t,i}-\theta_{t-1,i}\right|+c\right)^{\alpha-1}\left(\theta_{t+1,i}-\theta_i^*\right)^2 \\ = & \sum_{i=1}^{d}\frac{\Gamma(2-\alpha)}{2\eta}\left(\left|\theta_{1,i}-\theta_{0,i}\right|+c\right)^{\alpha-1}\left(\theta_{1,i}-\theta_i^*\right)^2 \\ & +\sum_{t=2}^{T}\sum_{i=1}^{d}\left[\frac{\Gamma(2-\alpha)}{2\eta_t}\left(\left|\theta_{t,i}-\theta_{t-1,i}\right|+c\right)^{\alpha-1}-\frac{\Gamma(2-\alpha)}{2\eta_{t-1}}\left(\left|\theta_{t-1,i}-\theta_{t-2,i}\right|+c\right)^{\alpha-1}\right]\cdot\left(\theta_{t,i}-\theta_i^*\right)^2\end{aligned} \tag{16}$$

将假设(3)代入(16)式中，可得：

$$\begin{aligned}\Delta_1 \le & \sum_{i=1}^{d}\frac{\Gamma(2-\alpha)\cdot D^2}{2\eta}\left(\left|\theta_{1,i}-\theta_{0,i}\right|+c\right)^{\alpha-1} \\ & +\sum_{t=2}^{T}\sum_{i=1}^{d}\left[\frac{\Gamma(2-\alpha)\cdot D^2}{2\eta_t}\left(\left|\theta_{t,i}-\theta_{t-1,i}\right|+c\right)^{\alpha-1}-\frac{\Gamma(2-\alpha)\cdot D^2}{2\eta_{t-1}}\left(\left|\theta_{t-1,i}-\theta_{t-2,i}\right|+c\right)^{\alpha-1}\right] \\ = & \sum_{i=1}^{d}\frac{\Gamma(2-\alpha)\cdot D^2\sqrt{T}}{2\eta}\left(\left|\theta_{T,i}-\theta_{T-1,i}\right|+c\right)^{\alpha-1}\end{aligned} \tag{17}$$

当 $0<\alpha<1$ 时，$\left(\left|\theta_{T,i}-\theta_{T-1,i}\right|+c\right)^{\alpha-1}\le c^{\alpha-1}$，

当 $1\le\alpha<2$ 时，$\left(\left|\theta_{T,i}-\theta_{T-1,i}\right|+c\right)^{\alpha-1}\le(2D+c)^{\alpha-1}$，

定义 $M\triangleq\max\left\{c^{\alpha-1},(2D+c)^{\alpha-1}\right\}$，则 $\left(\left|\theta_{T,i}-\theta_{T-1,i}\right|+c\right)^{\alpha-1}\le M$，因此：

$$\begin{aligned}\Delta_1 & \le \sum_{i=1}^{d}\frac{\Gamma(2-\alpha)\cdot D^2\sqrt{T}}{2\eta}\left(\left|\theta_{T,i}-\theta_{T-1,i}\right|+c\right)^{\alpha-1} \\ & \le \frac{\Gamma(2-\alpha)\cdot D^2 dM\sqrt{T}}{2\eta}\end{aligned} \tag{18}$$

到此为止就完成了第一部分的处理，对于第二部分，

由于梯度裁剪，$\left|{}_{\theta_{t-1,i}}D^{\alpha}_{\theta_{t,i}}L(\theta_t)\right|\le 1$，有：

$$\left[{}_{\theta_{t-1,i}}D^{\alpha}_{\theta_{t,i}}L(\theta_t)\right]^2=\left|{}_{\theta_{t-1,i}}D^{\alpha}_{\theta_{t,i}}L(\theta_t)\right|\cdot\left|{}_{\theta_{t-1,i}}D^{\alpha}_{\theta_{t,i}}L(\theta_t)\right|\le\left|{}_{\theta_{t-1,i}}D^{\alpha}_{\theta_{t,i}}L(\theta_t)\right| \tag{19}$$

因此，

$$\begin{aligned}\Delta_2 & =\sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\eta_t\Gamma(2-\alpha)}{2}\left(\left|\theta_{t,i}-\theta_{t-1,i}\right|+c\right)^{\alpha-1}\left[{}_{\theta_{t-1,i}}D^{\alpha}_{\theta_{t,i}}L(\theta_t)\right]^2 \\ & \le\frac{\eta M\Gamma(2-\alpha)}{2}\sum_{t=1}^{T}\sum_{i=1}^{d}\frac{1}{\sqrt{t}}\cdot\left[{}_{\theta_{t-1,i}}D^{\alpha}_{\theta_{t,i}}L(\theta_t)\right]^2 \\ & \le\frac{\eta M\Gamma(2-\alpha)}{2}\sum_{i=1}^{d}\sum_{t=1}^{T}\frac{1}{\sqrt{t}}\cdot\left|{}_{\theta_{t-1,i}}D^{\alpha}_{\theta_{t,i}}L(\theta_t)\right|\end{aligned} \tag{20}$$

由于

$$\begin{aligned}\sum_{t=1}^{T}\frac{1}{\sqrt{t}}\cdot\left|{}_{\theta_{t-1,i}}D^{\alpha}_{\theta_{t,i}}L(\theta_t)\right| & \le\sqrt{\sum_{t=1}^{T}\left(\frac{1}{\sqrt{t}}\right)^2}\cdot\sqrt{\sum_{t=1}^{T}\left|{}_{\theta_{t-1,i}}D^{\alpha}_{\theta_{t,i}}L(\theta_t)\right|^2} \\ & \le\sqrt{1+\ln T}\cdot D^{\alpha}_{\theta_{1,i}:\theta_{T,i}}L\end{aligned} \tag{21}$$

其中，

$$\begin{aligned}\sum_{i=1}^{d} D_{\theta_{1,i}:\theta_{T,i}}^{\alpha} L &\le \sqrt{d}\cdot\sqrt{\sum_{i=1}^{d}\left(D_{\theta_{1,i}:\theta_{T,i}}^{\alpha} L\right)^2} \\ &= \sqrt{d}\cdot\sqrt{\sum_{t=1}^{T}\left\|{}_{\theta_{t-1}} D_{\theta_t}^{\alpha} L\left(\theta_t\right)\right\|^2} \\ &\le \sqrt{d}\cdot\sqrt{T\cdot G^2} \\ &= \sqrt{d}G\cdot\sqrt{T}\end{aligned} \tag{22}$$

因此可以得到：

$$\Delta_2 \le \frac{\sqrt{d}G\eta M\Gamma\left(2-\alpha\right)}{2}\sqrt{1+\ln T}\cdot\sqrt{T} \tag{23}$$

综合以上讨论，本文对遗憾函数做出如下估计：

$$\begin{aligned}\frac{R(T)}{T} &\le \frac{\frac{\Gamma\left(2-\alpha\right)\cdot D^2 dM}{2\eta}\sqrt{T}+\frac{\sqrt{d}G\eta M\Gamma\left(2-\alpha\right)}{2}\sqrt{1+\ln T}\cdot\sqrt{T}}{T} \\ &= \frac{\Gamma\left(2-\alpha\right)\cdot D^2 dM}{2\eta}\cdot\frac{1}{\sqrt{T}}+\frac{\sqrt{d}G\eta M\Gamma\left(2-\alpha\right)}{2}\cdot\frac{\sqrt{1+\ln T}}{\sqrt{T}} \\ &\to 0\end{aligned} \tag{24}$$

根据(11)可知，分数阶梯度下降法具有次线性的遗憾。

4. 数值实验

4.1. 数据集

本文采用的数据集为 CIFAR-10，是由 Alex Krizhevshy 和 IIya Sutskever 整理的一个用于识别现实物体的小型数据集。CIFAR-10 中的每张图像都是 32 × 32 尺寸的三通道彩色图片，共有 10 个类别，每个类别分别拥有 5000 张训练图像以及 1000 张测试图像，所以整个数据集共有 60,000 张图像用于模型训练及测试。由于该数据集中的图像均源于现实中的真实物体，不仅图像噪声大，而且它们之间的特征差别很大，这为图像识别带来了巨大的困难，因此 CIFAR-10 成为了检验新模型与新算法的基准数据集。

4.2. 网络结构

卷积神经网络是专门处理与图像相关任务的深度网络结构，ResNet 正是 CNN 的一种，它通过添加快捷连接解决了深层网络难以训练的问题。本文采用的卷积神经网络模型为 ResNet18，它的网络深度为 18，共包含了 17 个卷积层和 1 个全连接层，第一个卷积层将图像的通道拓展到 64 通道，余下的 16 个卷积层组成了 4 个残差块，每个残差块有 4 个卷积层，第一个残差块保持输入输出图像尺寸不变，而其它 3 个残差块则会使输入图像通道数翻倍，尺寸减半，最后的全连接层用于预测图像的类别。为了使得每层输出与输入的方差保持一致，网络参数均采用 Xavier 初始化。

4.3. 超参数设置

本实验的主要超参数有分数阶梯度算法的阶次 α，学习率 η，常数 c 以及训练轮次 T。每次实验中，常数及轮次均被设置为固定值，即 $c=0.01$，$T=100$，α 和 η 取非固定值，这可以体现阶次对算法的影响以及算法对学习率的敏感程度。

4.4. 结果分析

表 1 和表 2 展示了学习率设置为 0.02、0.1、0.5、1.0 和 1.5 的情况下不同阶次的算法的运行结果，从表中可以看出：

1) 当学习率较小时，一般来说，随着阶次的增大，训练集的误差也在不断减小，而当学习率较大时，较大的阶次则会导致学习过程的不稳定。因此较小的学习率匹配更高的阶次，较大的学习率匹配较小的阶次

2) 学习率为 0.02 时，普通的梯度下降法达到了 85.19%的训练集精度和 80.14%的测试集精度，在设置阶次为 1.8 的情况下，两者分别上升到了 98.514%和 87.92%。学习率为 0.1 时，普通的梯度下降法达到了 98.016%的训练集精度和 88.22%的测试集精度，在设置阶次为 1.9 的情况下，两者分别上升到了 99.562%和 90.04%。当学习率较大时，虽然分数阶算法的优势有所下降，但是最优的分数阶算法仍在学习率取 1.0 和 1.5 的情况下，相较于普通的梯度算法分别提升了 1.36%和 0.54%的测试集精度，这说明分数阶梯度下降法在一定条件下拥有更强的搜索能力，可以提高模型的拟合程度和泛化能力。

Table 1. Training accuracy obtained by fractional gradient descent algorithm with different orders and learning rates
表 1. 不同阶次的算法在选择不同学习率时的训练集精度

阶次\学习率	0.02	0.1	0.5	1.0	1.5
0.1	0.39996	0.60808	0.87472	0.98482	0.9955
0.2	0.46266	0.65512	0.9098	0.98908	0.99762
0.3	0.49604	0.71704	0.95868	0.9955	0.99834
0.4	0.57444	0.77286	0.9717	0.99672	0.99846
0.5	0.62204	0.82022	0.97978	0.99862	0.9977
0.6	0.6735	0.86564	0.99578	0.99802	0.99902
0.7	0.72134	0.90448	0.99762	0.99918	0.99894
0.8	0.77526	0.94428	0.99724	0.9965	0.99822
0.9	0.81652	0.96406	0.99686	0.99818	0.99662
1.0	0.8519	0.98016	0.99766	0.99746	0.99842
1.1	0.89412	0.98506	0.9969	0.9981	0.99768
1.2	0.92642	0.98676	0.99714	0.99738	0.99618
1.3	0.95388	0.98484	0.99584	0.9968	0.98304
1.4	0.9639	0.99278	0.98946	0.09914	0.09566
1.5	0.97408	0.97974	0.98864	0.98592	0.9769
1.6	0.98168	0.992	0.98996	0.0976	0.09874
1.7	0.9825	0.98948	0.91666	0.09726	0.09926
1.8	0.98514	0.99248	0.0974	0.09638	0.09644
1.9	0.97868	0.99562	0.99384	0.09754	0.09688

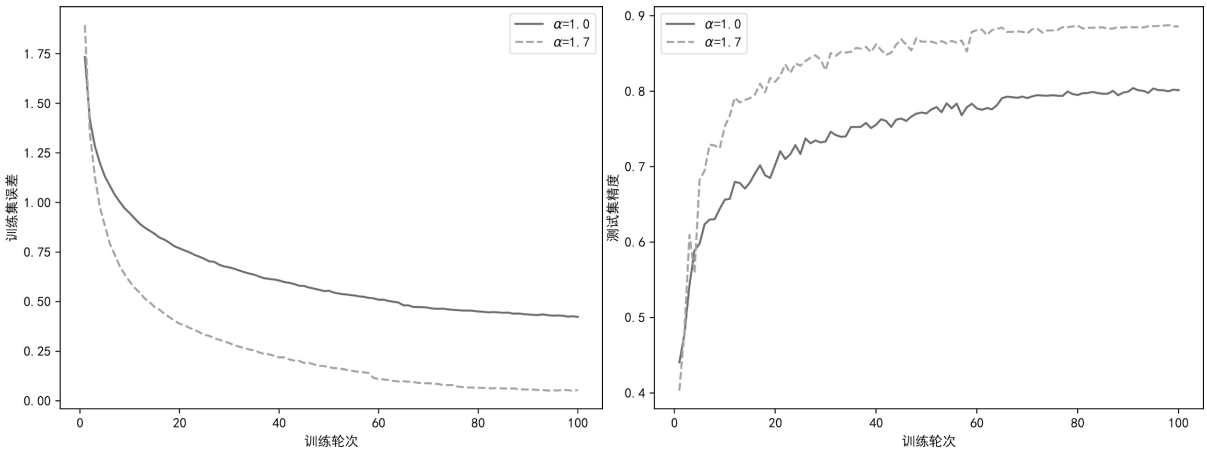
Table 2. Test accuracy obtained by fractional gradient descent algorithm with different orders and learning rates
表 2. 不同阶次的算法在选择不同学习率时的测试集精度

阶次\学习率	0.02	0.1	0.5	1.0	1.5
0.1	0.4032	0.6084	0.8082	0.9074	0.9186

续表

0.2	0.4669	0.6533	0.8352	0.9042	0.9227
0.3	0.4959	0.7017	0.8684	0.9151	0.9249
0.4	0.5771	0.7425	0.8768	0.9223	0.9203
0.5	0.623	0.7793	0.8796	0.9252	0.9276
0.6	0.669	0.8123	0.9048	0.9259	0.9297
0.7	0.7114	0.8254	0.9048	0.9242	0.9292
0.8	0.7442	0.8587	0.9053	0.9107	0.9337
0.9	0.7774	0.8681	0.9086	0.9082	0.9202
1.0	0.8014	0.8822	0.9094	0.9123	0.9283
1.1	0.8272	0.883	0.9034	0.9178	0.9169
1.2	0.8528	0.866	0.8994	0.9176	0.9092
1.3	0.8668	0.8764	0.9028	0.9114	0.9052
1.4	0.8729	0.8885	0.8918	0.1	0.1
1.5	0.8753	0.8613	0.8808	0.8864	0.8717
1.6	0.8739	0.8924	0.8955	0.1	0.1
1.7	0.8855	0.8898	0.8294	0.1	0.1
1.8	0.8792	0.8853	0.1	0.1	0.1
1.9	0.8696	0.9004	0.8802	0.1	0.1

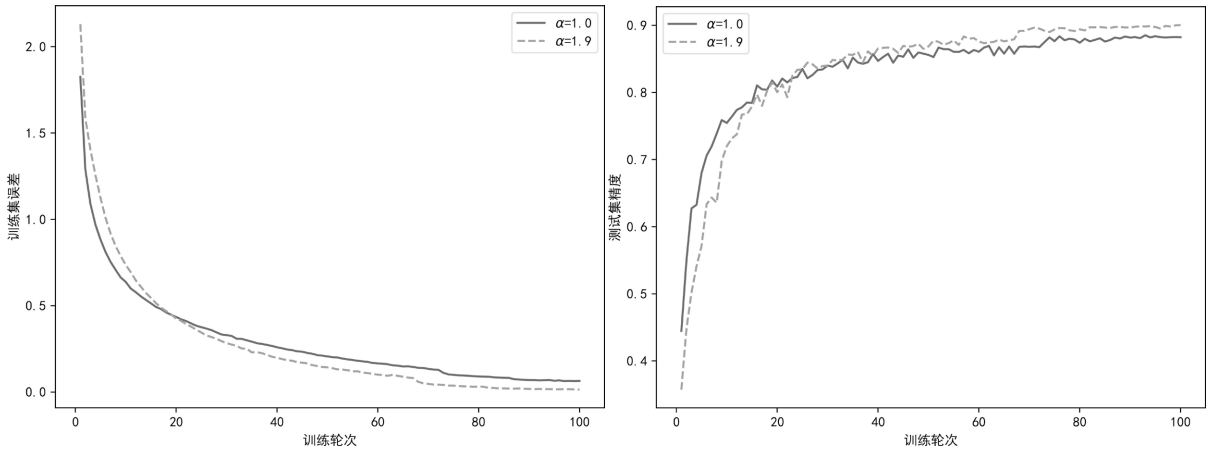
图 1~3 给出了取不同学习率时整数阶梯度下降算法与最优的分数阶梯度下降算法的训练集误差和测试集精度的对比，注意到当$\alpha=1$时分数阶梯度下降算法退化为整数阶梯度下降算法。从图 1 和图 2 可以看出，当学习率较小时，虽然在前期训练中分数阶算法的表现可能逊于整数阶算法，但是在后期训练中分数阶算法的性能便超越了整数阶算法，在整个训练过程中表现出了更快的收敛速度，并且取得了更高的精度。当学习率较大时，图 3 表明虽然普通的梯度下降法在收敛速度方面与分数阶算法表现相当，但是在选择合适阶次的情况下，分数阶算法往往能取得更高的测试集精度，提高模型的实际应用价值。



(a) (b)

Figure 1. Comparison between integral gradient method and fractional gradient method with alpha = 1.7 when learning rate = 0.02

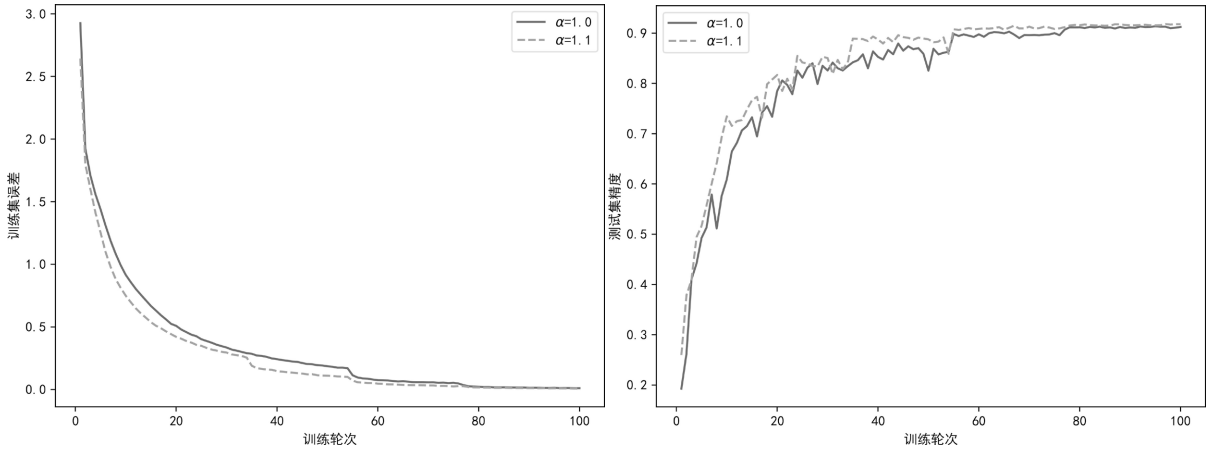
图 1. 学习率为 0.02 时整数阶梯度方法与阶次为 1.7 的分数阶梯度方法的对比



(a) (b)

Figure 2. Comparison between integral gradient method and fractional gradient method with alpha = 1.9 when learning rate = 0.1

图 2. 学习率为 0.1 时整数阶梯度方法与阶次为 1.9 的分数阶梯度方法的对比



(a) (b)

Figure 3. Comparison between integral gradient method and fractional gradient method with alpha = 1.1 when learning rate = 1.0

图 3. 学习率为 1.0 时整数阶梯度方法与阶次为 1.1 的分数阶梯度方法的对比

5. 总结

本文从提出更适用于神经网络训练的优化算法出发，以 Caputo 分数阶微分为基础构造了新型的分数阶梯度下降法。该方法改变了积分下界，将分数阶阶次拓展到了(0, 2)区间，扩大了该算法的优化空间。在理论方面，本文结合了梯度裁剪机制，从遗憾函数的角度详细地证明了该算法具有次线性的遗憾，保证了算法的理论可行性。最后本文通过数值实验，表明在选择合适阶次的情况下，分数阶梯度下降法在收敛精度和收敛速度方面均具有更加良好的表现。

参考文献

[1] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W. and Jackel, L. (1989) Handwritten Digit Recognition with a Back-Propagation Network. Advances in Neural Information Processing Systems, **2**, 396-404.

[2] Jordan, M.I. (1997) Serial Order: A Parallel Distributed Processing Approach. *In Advances in Psychology*, **121**,

471-495.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017) Attention Is All You Need. *Advances in Neural Information Processing Systems*, **30**, 6000-6010.

[4] He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/cvpr.2016.90

[5] Tan, Y., He, Z. and Tian, B. (2015) A Novel Generalization of Modified LMS Algorithm to Fractional Order. *IEEE Signal Processing Letters*, **22**, 1244-1248. https://doi.org/10.1109/lsp.2015.2394301

[6] Khan, S., Naseem, I., Malik, M.A., Togneri, R. and Bennamoun, M. (2018) A Fractional Gradient Descent-Based RBF Neural Network. *Circuits, Systems, and Signal Processing*, **37**, 5311-5332. https://doi.org/10.1007/s00034-018-0835-3

[7] Zhou, X., Zhao, C. and Huang, Y. (2023) A Deep Learning Optimizer Based on Grünwald—Letnikov Fractional Order Definition. *Mathematics*, 11, Article 316. https://doi.org/10.3390/math11020316

[8] Chaudhary, N.I., Raja, M.A.Z., Khan, Z.A., Mehmood, A. and Shah, S.M. (2022) Design of Fractional Hierarchical Gradient Descent Algorithm for Parameter Estimation of Nonlinear Control Autoregressive Systems. Chaos, *Solitons & Fractals*, **157**, Article ID: 111913. https://doi.org/10.1016/j.chaos.2022.111913

[9] Liu, J., Zhai, R., Liu, Y., Li, W., Wang, B. and Huang, L. (2021) A Quasi Fractional Order Gradient Descent Method with Adaptive Stepsize and Its Application in System Identification. *Applied Mathematics and Computation*, **393**, 125797. https://doi.org/10.1016/j.amc.2020.125797

[10] Pu, Y., Zhou, J., Zhang, Y., Zhang, N., Huang, G. and Siarry, P. (2015) Fractional Extreme Value Adaptive Training Method: Fractional Steepest Descent Approach. *IEEE Transactions on Neural Networks and Learning Systems*, **26**, 653-662. https://doi.org/10.1109/tnnls.2013.2286175

[11] Chen, Y., Gao, Q., Wei, Y. and Wang, Y. (2017) Study on Fractional Order Gradient Methods. *Applied Mathematics and Computation*, **314**, 310-321.

[12] Wei, Y., Kang, Y., Yin, W. and Wang, Y. (2020) Generalization of the Gradient Method with Fractional Order Gradient Direction. *Journal of the Franklin Institute*, **357**, 2514-2532.

[13] Zhang, H., Pu, Y., Xie, X., Zhang, B., Wang, J. and Huang, T. (2021) A Global Neural Network Learning Machine: Coupled Integer and Fractional Calculus Operator with an Adaptive Learning Scheme. *Neural Networks*, **143**, 386-399. https://doi.org/10.1016/j.neunet.2021.06.021

[14] Cesa-Bianchi, N., Lugosi, G. and Stoltz, G. (2006) Regret Minimization under Partial Monitoring. *Mathematics of Operations Research*, **31**, 562-580. https://doi.org/10.1287/moor.1060.0206