计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(7):220−224 [doi: 10.15888/j.cnki.csa.007979]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于差分修正的 SGDM 算法①

袁 炜, 胡 飞

(天津大学 数学学院, 天津 300350)
通讯作者: 胡 飞, E-mail: hfzdzy@126.com

摘 要: 当前, 应用广泛的一阶深度学习优化器包括学习率非自适应优化器和学习率自适应优化器, 前者以 SGDM 为代表, 后者以 Adam 为代表, 这两类方法都使用指数滑动平均法来估计总体的梯度. 然而使用指数滑动平均法来估计总体梯度是有偏差且具有滞后性的, 本文提出基于差分修正的 SGDM 算法——RSGDM 算法. 我们的贡献主要有 3 点: 1) 分析 SGDM 算法里指数滑动平均法带来的偏差和滞后性. 2) 使用差分估计项来修正 SGDM 算法里的偏差和滞后性, 提出 RSGDM 算法. 3) 在 CIFAR-10 和 CIFAR-100 数据集上实验证明了在收敛精度上我们的 RSGDM 算法比 SGDM 算法更优.

关键词: 深度学习; 一阶优化器; SGDM 算法; 差分

引用格式: 袁炜,胡飞.基于差分修正的 SGDM 算法.计算机系统应用,2021,30(7):220–224. http://www.c-s-a.org.cn/1003-3254/7979.html

Rectified SGDM Algorithm Based on Difference

YUAN Wei, HU Fei

(School of Mathematics, Tianjin University, Tianjin 300350, China)

Abstract: Currently, the widely used first-order deep learning optimizers include non-adaptive learning rate optimizers such as SGDM and adaptive learning rate optimizers like Adam, both of which estimate the overall gradient through exponential moving average. However, such a method is biased and hysteretic. In this study, we propose a rectified SGDM algorithm based on difference, i.e. RSGDM. Our contributions are as follows: 1) We analyze the bias and hysteresis triggered by exponential moving average in the SGDM algorithm. 2) We use the difference estimation term to correct the bias and hysteresis in the SGDM algorithm, and propose the RSGDM algorithm. 3) The experiments on CIFAR-10 and CIFAR-100 datasets proves that our RSGDM algorithm is higher than the SGDM algorithm in convergence accuracy.

Key words: deep learning; first order optimization; SGDM algorithm; difference

当前, 深度神经网络模型在视觉[1]、文本[2] 和语音[3] 等任务上都有着非常好的表现. 但是随着神经网络层数的加深, 模型的训练变得越来越困难. 因此, 研究人员开始关注优化方法. 随机梯度下降法 (SGD) 是求解优化问题的一种简单有效的方法, SGD 被广泛应用于实际问题[4]. 该方法根据小批量 (mini-batch) 样本的可微损失函数的负梯度方向对模型参数进行更新, SGD 具有训练速度快、精度高的优点. 但由于其参数是每次按照一个小批量样本更新的, 因此如果每个小批量样本的特性差异较大, 更新方向可能会发生较大的变化, 这导致了它不能快速收敛到最优解的问题. 因此, SGD 有很多变种, 一般可以分为两类. 第一类是学习率非自适应性方法, 使用梯度的一阶矩估计——SGDM[5], 结合每个小批量样本的梯度求滑动平均值来更新参数,

① 收稿时间: 2020-10-20; 修改时间: 2020-11-18, 2020-11-24, 2020-12-01; 采用时间: 2020-12-08; csa 在线出版时间: 2021-06-30

极大地解决了 SGD 算法收敛速度慢的问题, 这个方法目前应用非常广泛, 本文就针对这个方法进行改进. 第二类是学习率自适应性方法, 利用梯度的二阶矩估计实现学习率的自适应调整, 包括 AdaGrad[6], RMSProp[7], AdaDelta[8], Adam[9]. 其中 Adam 是深度学习中最常见的优化算法, 虽然 Adam 在训练集上的收敛速度相对较快, 但在训练集上的收敛精度往往不如非自适应性优化方法 SGDM, 且泛化能力不如 SGDM. AmsGrad[10] 是对 Adam 的一个重要改进, 但是最近的研究表明, 它并没有改变自适应优化方法的缺点, 实际效果也没有太大改善. RAdam[11] 是 Adam 的一个新变体, 可以自适应的纠正自适应学习率的变化.

目前很多研究都集中在第二类学习率自适应性方法上, 却忽略了最基本的问题. 无论自适应还是非自适应方法都用了指数滑动平均法, 这些方法都试图用指数滑动平均得到近似样本总体的梯度, 然而这种方法是有偏且具有滞后性的. 针对这一问题我们提出了 RSGDM 算法, 我们的方法计算梯度的差分, 即计算当前迭代梯度与上一次迭代梯度的差, 相当于梯度的变化量, 我们在每次迭代时使用指数滑动平均估计当前梯度的变化量, 然后用这一项与当前梯度的估计值进行加权求和, 这样的做法可以降低偏差且缓解滞后性.

1 RSGDM 算法

算法 1. RSGDM 算法: 所有向量计算都是对应元素计算

输入: 学习率α, 指数衰减率β, 随机目标函数$f(\theta)$

随机初始化参数向量θ_0, 初始化参数向量m_0=0,z_0=0,Δg_1=0, 初始化时间步长t=0

While θ_t不收敛 **do**

$t\leftarrow t+1$

计算 t 时间步长上随机目标函数的梯度:

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

计算差分:

$\Delta g_t=g_t-g_{t-1}$

更新梯度和差分的一阶矩估计:

$m_t\leftarrow\beta*m_{t-1}+(1-\beta)*g_t$

$z_t\leftarrow\beta*z_{t-1}+(1-\beta)*\Delta g_t$

用差分的一阶矩估计修正对梯度的估计:

$n_t\leftarrow m_t+\beta*z_t$

更新参数:

$\theta_t\leftarrow\theta_{t-1}-\alpha*n_t$

end While

输出: 结果参数θ_t

算法 1 展示了 RSGDM 算法的整体流程. 设$f(\theta)$为关于θ可微的目标函数, 我们希望最小化这个目标函数关于参数θ的期望值, 即最小化$E[f(\theta)]$. 我们使用$f_1(\theta),f_2(\theta),\cdots,f_T(\theta)$来表示时间步长$1,2,\cdots,T$对应的随机目标函数. 这个随机性来自于每个小批量的随机采样 (mini-batch) 或者函数固有的噪声. 梯度为$g_t=\nabla_\theta f_t(\theta)$, 即在时间步长$t$时, 目标函数$f_t$关于参数$\theta$的偏导向量. 使用$\Delta g_t=g_t-g_{t-1}$表示时间步长$t$和时间步长$t-1$对应的梯度差, 即本文提到的差分. 不同于 SGDM 算法只对g_t做指数滑动平均, RSGDM 算法对g_t和Δg_t都做指数滑动平均, 并用后者的滑动平均值修正前者, 下面列出 SGDM 算法和 RSGDM 算法的更新公式.

1) SGDM 算法:

$$m_t=\beta*m_{t-1}+(1-\beta)*g_t \tag{1}$$
$$\theta_t=\theta_{t-1}-\alpha*m_t$$

2) RSGDM 算法:

$$m_t=\beta*m_{t-1}+(1-\beta)*g_t$$
$$z_t=\beta*z_{t-1}+(1-\beta)*\Delta g_t \tag{2}$$
$$n_t=m_t+\beta*z_t \tag{3}$$
$$\theta_t=\theta_{t-1}-\alpha*n_t$$

可以看出, RSGDM 算法比 SGDM 算法多出式 (2) 和式 (3), 本文第 2 节将证明 SGDM 中的式 (1) m_t对总体g_t的估计是有偏的, 我们就是使用式 (2) 和式 (3) 来进行偏差修正的. RSGDM 算法相比于 SGDM 算法没有增加多余的超参, 不会增加我们训练模型调参的负担.

2 偏差及滞后性分析

首先我们分析用指数滑动平均估计总体梯度的偏差, 首先由 SGDM 算法中的式 (1) 可以得到:

$$m_t=(1-\beta)*\sum_{i=1}^{t}\beta^{t-i}*g_i \tag{4}$$

对式 (4) 两边取期望可得:

$$\begin{aligned}E(m_t)&=(1-\beta)*E\left(\sum_{i}^{t}\beta^{t-i}*g_i\right)=(1-\beta)*\left[\beta^{t-1}*E(g_1)\right.\\&\left.+\beta^{t-2}*E(g_2)+\cdots+\beta*E(g_{t-1})+E(g_t)\right]\\&=\left(1-\beta^t\right)*E(g_t)+(1-\beta)*\xi\end{aligned} \tag{5}$$

可以发现$E(m_t)\neq E(g_t)$, 其中:

$$\begin{aligned}\xi&=\sum_{i=1}^{t}\beta^{t-i}[E(g_i)-E(g_t)]=\beta^{t-1}*[g_1-E(g_t)]\\&+\beta^{t-2}*[g_2-E(g_t)]+\cdots+\beta*[g_{t-1}-E(g_t)]\end{aligned} \tag{6}$$

当迭代次数较多时, $1-\beta^t$可以忽略不计, 最大的

偏差就来自于式 (6). 如果g_t是一个平稳序列, 即$E(g_t)$ = C(C 是常数) 时, 有$\xi = 0$, 此时m_t是g_t的一个无偏估计. 但实际情况下, 显然这是不可能的, 所以m_t是g_t的有偏估计, 且偏差主要来自于ξ. 并且这个偏差会导致滞后性, 比如梯度如果一直处在增大的状态, 由于前面历史时刻梯度值较小也会导致估计值m_t偏小一些, 或者梯度如果一直增大, 但从某个时刻开始减小, 由于历史梯度的影响梯度的估计值m_t可能还没有反应过来, 这就是本文所说的滞后性带来的影响. 针对这种情况我们提出了 RSGDM 算法, 使用梯度的差分 (变化量) 估计来修正梯度的估计值m_t. 直观上可以这样理解, 如果梯度越来越大且差分的估计也是大于 0 的, 那么这个修正项起到加速收敛的作用, 如果梯度越来越大, 在某一时刻要开始减小, 那这个修正项就会起到修正梯度下降方向的作用. 下面我们从公式上解释 RSGDM 算法的优势:

首先由式 (2), 我们可以得到:

$$z_t = (1-\beta) * \sum_{i=2}^{t} \beta^{t-i} * \Delta g_i \tag{7}$$

对 RSGDM 算法中的式 (3) 两边取期望, 我们可以得到:

$$E(n_t) = \left(1-\beta^t\right) * E(g_t) + (1-\beta) * \xi + \beta(1-\beta) * \left(\sum_{i=2}^{t} \beta^{t-i} * \Delta g_i\right) \tag{8}$$

n_t是修正后的对g_t的估计, 我们可以对比式 (5) 和式 (8), 我们设$\varsigma = \xi + \beta\left(\sum_{i=2}^{t} \beta^{t-i} * \Delta g_i\right)$, 不难发现$\varsigma$是 RSGDM 算法的主要偏差来源, 我们展开$\varsigma$, 得到:

$$\varsigma = \sum_{i=1}^{t-2} \beta^{t-i} * [g_{i+1} - E(g_t)] = \beta^{t-1} * [g_2 - E(g_t)] + \beta^{t-2} * [g_3 - E(g_t)] + \cdots + \beta^2 * [g_{t-1} - E(g_t)] \tag{9}$$

我们对比式 (6)SGDM 算法的偏差项ξ和式 (9) RSGDM 算法的偏差项ς, 可以看出ξ受历史梯度$g_1, g_2, \cdots, g_{t-1}$影响, 而 ς受历史梯度 $g_2, g_3, \cdots, g_{t-1}$影响. 由于$t$很大且$\beta$ 小于 0, $\beta^{t-1} * [g_1 - E(g_t)]$接近于 0 可以忽略, 那么可以得到$\varsigma = \beta * \xi$. 可以看出 RSGDM 算法偏差项$\varsigma$少了历史梯度$g_1$的影响, 且由于$\beta$小于 0, 所以$|\varsigma| \le |\xi|$. 综上, 我们可以得出 RSGDM 算法对比 SGDM 算法有更小的偏差, 且受历史梯度影响要小 (缓解了滞后性).

3 实验

本节我们通过实验证明我们的 RSGDM 算法比 SGDM 算法更有优势. 我们选择图像分类任务来验证算法的优越性, 实验使用了 CIFAR-10 和 CIFAR-100 数据集[12]. CIFAR-10 数据集和 CIFAR-100 数据集均由 32×32 分辨率的 RGB 图像组成, 其中训练集均是 50 000 张图片, 测试集 10 000 张图片. 我们在 CIFAR-10 数据集上进行 10 种类别的分类, 在 CIFAR-100 数据集上进行 100 种类别的分类. 我们分别使用 ResNet18 模型和 ResNet50[13] 模型在 CIFAR-10 和 CIFAR-100 数据集上进行图像分类任务, 每个任务使用 SGDM 算法和我们的 RSGDM 算法进行比较, 评价的指标为分类的准确率. 我们使用的深度学习框架是 PyTorch, 训练的硬件环境为单卡 NVIDIA RTX 2080Ti GPU. 实验中的批量大小 (batch-size) 设置为 128, SGDM 算法和 RSGDM 算法的两个超参数设置一样, 其中动量β= 0.9, 初始学习率α= 0.01, 训练时我们使用了权重衰减的方法来防止过拟合, 衰减参数设置为 5e–4, 学习率每 50 轮 (epoch) 减少一半.

表 1 和表 2 分别给出了 ResNet18 和 ResNet50 使用不同的优化器在 CIFAR-10 和 CIFAR-100 数据集上图像分类的准确率. 我们可以看到在 CIFAR-10 数据集上, SGDM 和 RSGDM 训练精度都达到了 100%, 测试精度我们的 RSGDM 比 SGDM 高了 0.14%. 在 CIFAR-100 数据集上, SGDM 和 RSGDM 训练精度都是 99.98%, 在测试精度上 RSGDM 比 SGDM 高了 0.57%.

表 1 ResNet18 在 CIFAR-10 上分类准确率 (%)

算法	训练集	测试集
SGDM算法	100	94.48
RSGDM算法	100	94.62

表 2 ResNet50 在 CIFAR-100 上分类准确率 (%)

算法	训练集	测试集
SGDM算法	99.98	76.70
RSGDM算法	99.98	77.27

图 1–图 4 给出了 ResNet18 在 CIFAR-10 上使用 SGDM 算法和 RSGDM 算法的实验结果, 包括训练准确率、训练损失、测试准确率、测试损失. 由于我们实验设置每 50 个 epoch 学习率减半, 很明显可以看出 4 张图在 epoch50、100、150 均有一些波动. 总体上看在训练准确率和训练损失上, 两种方法无论收敛速度和收敛精度都大体相同, 在收敛精度上训练后期我们的 RSGDM 算法更具优势.

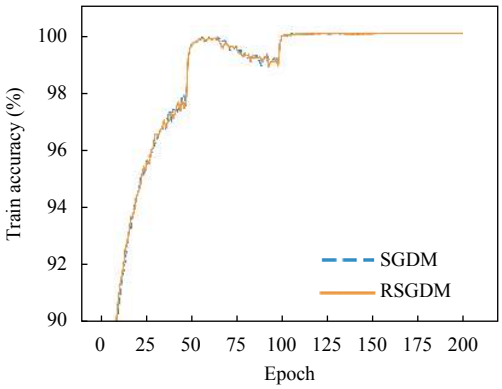

图 1 ResNet18 在 CIFAR-10 上的训练准确率

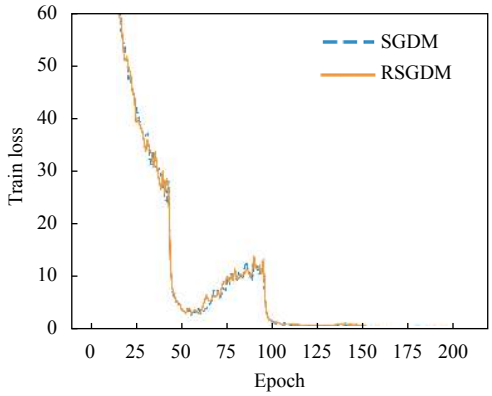

图 2 ResNet18 在 CIFAR-10 上的训练损失

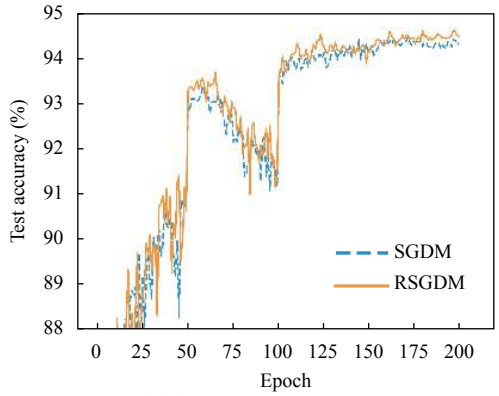

图 3 ResNet18 在 CIFAR-10 上的测试准确率

图 5–图 8 给出了 ResNet50 在 CIFAR-100 上使用 RSGDM 算法和 SGDM 算法的实验结果. 可以得出与 CIFAR-10 类似的结果, 在训练准确率和训练损失上二者的几乎相同, 但是在收敛精度上, RSGDM 算法的表现在这个数据集上要领先 SGDM 算法很多, 可以看测试准确率那张图, 从 100 个 epoch 之后, 我们的 RSGDM 方法始终准确率高于 SGDM, 并且最终准确率比 SGDM 高了 0.57%. 这进一步说明我们的方法是有效的.

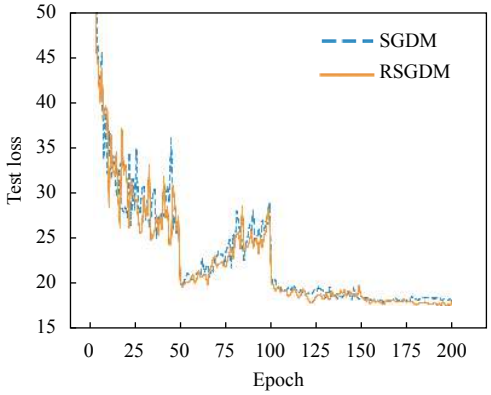

图 4 ResNet18 在 CIFAR-10 上的测试损失

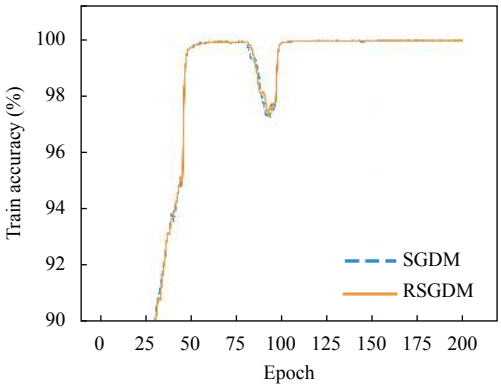

图 5 ResNet50 在 CIFAR-100 上的训练准确率

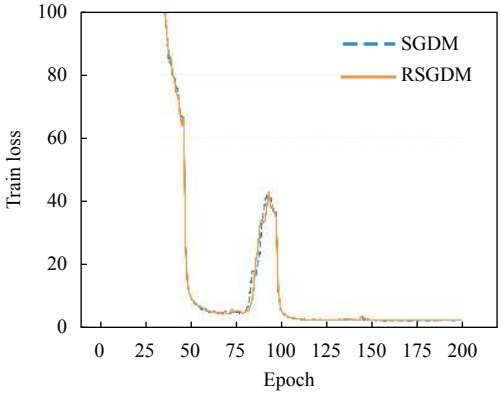

图 6 ResNet50 在 CIFAR-100 上的训练损失

4 结论

随着深度学习模型的不断复杂化, 训练一个精度高的模型越来越不容易, 一个好的优化器具有非常重要的作用. 本文分析了传统方法 SGDM 算法使用指数

滑动平均估计总体梯度带来的偏差与滞后性, 并理论上证明了这个偏差与滞后性存在. 我们提出了基于差分修正的 RSGDM 算法, 该算法对梯度的差分进行估计并使用这个估计项修正指数滑动平均对梯度的估计, 我们从理论上说明了算法的可行性, 并且在 CIFAR-10 和 CIFAR-100 两个数据集上进行图像分类任务的实验, 实验结果也进一步说明了我们的 RSGDM 算法的优势. 值得一提的是, 我们的 RSGDM 算法没有引入更多的需要调试的参数, 在不加重深度学习研究员调试超参负担的情况下, 提升了收敛精度. 本文开头介绍了当前这个领域的其他方法, 包括现在最常用的学习率自适应性算法 Adam 等, 未来我们会使用现在的改进思路去提出新的学习率自适应性算法, 提出更加准确高效的算法.

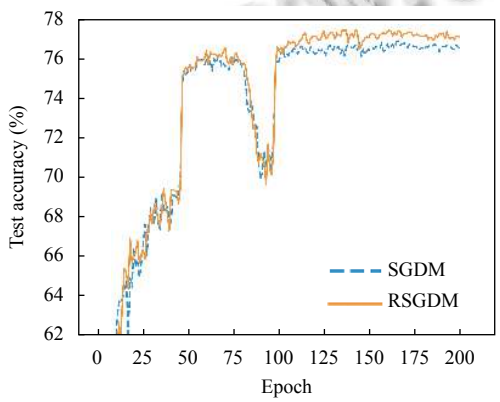

图 7 ResNet50 在 CIFAR-100 上的测试准确率

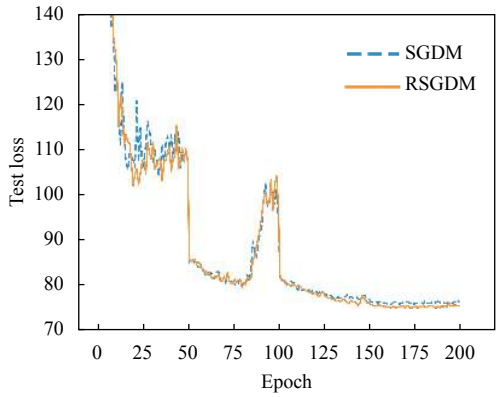

图 8 ResNet50 在 CIFAR-100 上的测试损失

参考文献

1 Ren SQ, He KM, Girshick R, *et al*. Faster R-CNN: Towards real-time object detection with region proposal networks. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal, QC, Canada. 2015. 91–99.

2 Cho K, van Merrienboer B, Gülçehre C, *et al*. Learning phrase representations using RNN encoder-decoder for statistical machine translation. Proceedings of the 2014 Conference on Empirocal Methods in Natural Language Processing (EMNLP). Doha, Qatar. 2014. 1724–1734.

3 Xiong W, Droppo J, Huang X, *et al*. Achieving human parity in conversational speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017, 25(12): 2410–2423.

4 Robbins H, Monro S. A stochastic approximation method. The Annals of Mathematical Statistics, 1951, 22: 400–407. [doi: 10.1214/aoms/1177729586]

5 Sutskever I, Martens J, Dahl G, *et al*. On the importance of initialization and momentum in deep learning. Proceedings of the 30th International Conference on International Conference on Machine Learning. Atlanta, GA, USA. 2013. 1139–1147.

6 Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 2011, 12: 2121–2159.

7 Tieleman T, Hinton G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012, 4(2): 26–31.

8 Zeiler MD. ADADELTA: An adaptive learning rate method. arXiv preprint arXiv: 12125701, 2012.

9 Kingma DP, Ba J. Adam: A method for stochastic optimization. Proceedings of the International Conference on Learning Representations. 2015.

10 Reddi SJ, Kale S, Kumar S. On the convergence of adam and beyond. Proceedings of the International Conference on Learning Representations. 2018.

11 Liu LY, Jiang HM, He PC, *et al*. On the variance of the adaptive learning rate and beyond. Proceedings of the International Conference on Learning Representations 2020. Addis Ababa, Ethiopia. 2020. 1–13.

12 Krizhevsky A. Learning multiple layers of features from tiny images. Toronto: University of Toronto, 2009.

13 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 770–778.