

文章编号: 1000-5862(2021)01-0060-07

基于 Fibonacci 采样的数据预处理方法研究

邓 泓^{1,3}, 刘志超², 彭莹琼^{1,3*}, 舒 晴¹, 何火娇¹

(1. 江西农业大学软件学院 江西 南昌 330045; 2. 江西农业大学计算机与信息工程学院 江西 南昌 330045;

3. 江西农业大学江西省农业信息技术重点实验室 江西 南昌 330045)

摘要: 提高神经网络(NN)的收敛速率和预测精度一直是人工智能领域的一个挑战性问题, 尽管许多研究人员已在研究中使用小批量数据训练神经网络获得了较好的效果, 但是这些方法并不够灵活. 针对这个问题, 该文提出了一种新的数据预处理算法即 Fibonacci 采样算法. 根据 Fibonacci 数列规则绘制一个新的训练数据序列, 这不仅可以恢复小批量数据的划分, 而且还可以提供更灵活的批量规模的选择. 实验结果表明: 在梯度下降之前, Fibonacci 数据划分序列能得到较好的实验结果. 在应用于相同的单个 CNN(5 层卷积神经网络)和 AlexNet 中, Fibonacci 采样算法比传统的小批量梯度下降算法能获得更高的准确度和更低的损失值, 并且在几种通用网络(LeNet、AlexNet、VGG-16、GoogLeNet)上的性能也取得显著提升.

关键词: 小批量; Fibonacci 采样; 神经网络

中图分类号: TP 389.1 **文献标志码:** A **DOI:** 10.16357/j.cnki.issn1000-5862.2021.01.09

0 引言

随机优化已在机器学习社区中得到了广泛研究^[1-6], 最早的随机优化工作可以追溯到 H. Robbins 等^[17]的研究. 对于随机问题的处理, 主要有 4 种解决方案: (i) 训练轮次减少; (ii) 小样本采样; (iii) 重要性采样; (iv) 通过“移位”减少差异. 通常在传统的随机优化步骤中, 每个步骤随机抽取在训练数据集中的样本进行训练, 然后根据最终损失结果反向回传并更新参数. 本文所提出 Fibonacci 采样算法采用的解决方案为小样本采样, 其中变量样本法研究^[18-20]也一直是研究的热点和难点, 由于样本大小的动态性质, 传统收敛结果需要附加条件进行样本均值的逼近. 如在已有文献研究中有许多用于在迭代 i 中选择坐标 j 的规则(包括循环规则^[21-24]、Gauss-Southwell 或其他基于贪婪定理的规则^[25-27]和随机规则^[28-33]). R. H. Byrd 等^[34]提出了一种精确尺寸的动态采样方法, 并将其应用于牛顿法. 杨杰明等^[35]利用基于密度分布的欠采样方法改善数据平衡, 使得最终预测精度提升. A. Shapiro 等^[36]使用

收敛指标来说明具体的类别问题以此提升预测精度. A. Shapiro 等^[37]在点集映射的框架中验证了动态采样的收敛结果, 并且 A. J. Kleywegt 等^[38]对此进行了进一步研究. 上述研究均在网络模型中额外增大了计算量的消耗. A. Shapiro 等^[39]基于类似策略对可变样本大小的方法建立模型, 为了计算新的样本大小, 在其策略中使用随机梯度预测的置信椭圆域. 然而, 由于样本协方差矩阵对于确定信息椭圆是必不可少的, 所以该方法在大规模数据计算上是不可行的. 因此, T. Homem-de-Mello^[40]提出了一种可变样本大小的方法, 该方法仅使用纯随机搜索方法概念下的函数值度量, 也就是说, 若样本满足相应的统计 t 检验, 则迭代继续确保算法可以确定良好的样本大小, 但是该算法在特定的样本环境下却无法进一步开展实验. 遵循着模拟路线, F. Bastin 等^[41]还提出了使用蒙特卡罗方法的不同样本量方法. 但是以上方法都在现有网络开销上额外增大了选择样本的消耗.

在现阶段的情况中研究人员通常会选择均匀采样定理, 而不是花费网络开销在训练样本量的选择上. 虽然均匀采样简化了分析, 但它会引入非常高的

收稿日期: 2020-03-16

基金项目: 国家自然科学基金(61363041), 江西省自然科学基金(20141BBF60051), 江西省教育厅科学技术研究(GJJ180234)和江西省教育厅科学技术研究(GJJ190208)资助项目.

作者简介: 邓 泓(1977-), 男, 江西都昌人, 副教授, 主要从事农业信息化与图像处理的研究. E-mail: jxaudh@aliyun.com

通信作者: 彭莹琼(1978-), 女, 江西萍乡人, 副教授, 主要从事计算机视觉处理的研究. E-mail: 29257353@qq.com

样本差异, 这会对最终优化结果的准确性产生负面影响. 比如随机梯度下降 (SGD) 等传统算法在整个学习过程中随机均匀地训练样本, 虽然随机梯度是真实梯度的无偏估计^[1-4]; 然而, 随机梯度估计的方差有可能非常高, 因为随机梯度可以在不同的样本中产生显著的变化. 在上述一系列的采样方法中, 动态采样遵循样本动态性但样本选择开销大, 均匀采样参照样本选择经验性但会导致样本差异大, 这 2 种采样均不能最有效地在采样方法上达成现阶段研究者实验所需, 为了解决这个问题, 本文提出了 Fibonacci 动态样本大小采样算法.

在研究了多个 Fibonacci 序列样本的随机优化之后, 本文提出了基于 Fibonacci 采样动态样本大小算法. 该算法的目标是有效地生成减少目标函数的方向, 从而减少了随机方差, 提高了分类预测精度. Fibonacci 采样算法是数据预处理的一个较好的实例, 在 LeNet、AlexNet 等网络模型中对 cifar-10 数据集的分类精度相较于传统均匀采样算法均取得了有效性的提升.

1 背景介绍

深度学习方面的研究在诸多领域中已经获得了很大的进展, 如视觉、文本分析等. SGD (随机梯度下降) 方法已经被证明为训练深度网络的优化方法, I. Sutskever 等^[42]、Qian Ning^[43] 和 D. P. Kingma 等^[44] 在 SGD 上进行了改进, 获得了良好的性能. SGD 优化网络损失参数 J , 用于最小化最终的训练损失. 参数 J 为

$$J = \arg \min_J \frac{1}{N} \sum_{i=1}^N L(x_i; J),$$

其中 $\{x_1, x_2, \dots, x_n\}$ 是训练数据集, N 表示每次参与训练的样本容量, 函数 L 代表每一轮的损失. 在 SGD 中, 训练分步进行, 本文的方法在每一步训练中使用样本容量为 q 的 $\{x_1, x_2, \dots, x_q\}$. 通过对于损失函数的训练批量规模参数的梯度近似, 得到计算公式为

$$\frac{1}{q} \frac{\partial L(x_i; J)}{\partial J}.$$

在通常情况下不同于 1 次抽取 1 个样本, 应用小批量的采样是非常有效的. (i) 关于小批量的梯度损失是对训练数据集梯度的预测; (ii) 得益于现代计算平台提供的并行性, 对小批量 q 的计算比整个训练数据更有效.

2 Fibonacci 采样算法在梯度下降中的运用

本文提出了一种基于 δ -规则的小批量优化算法, 其中在 Fibonacci 采样规则中使用训练样本的大小在算法的整个过程中是循环增加的. 在每次迭代时, 该方法选择在训练集中样本容量 $S_i \in [1, N]$ 的子集 S , 并将优化算法的步骤应用于如下目标函数:

$$J_S(w) = \sum_{i=1}^m \frac{1}{S_i} L(w; x_i),$$

其中 S_i 表示第 i 轮训练的样本容量, L 表示该轮训练的损失, w 为权重. 该方法的优势在于使用小的初始样本可以在早期阶段时快速收敛. 在该采样策略中的 2 个基本部分是触发样本量的 Fibonacci 增加和选择新样本的阈值条件. 在采样策略中样本量 S_i 等于 S_{i-1} 和 S_{i-2} 的总和, 并且非常接近 S_{i-1} 的 2 倍, 所以可以推出如下不等式, 该规则适用于绝大部分情况:

$$|\nabla J_{S_i}(w)| = \frac{1}{S_i} \sum_{k=1}^{S_i} L(w; x_k) < \frac{1}{S_{i-1}} \sum_{k=1}^{S_{i-1}} L(w; x_k) = |\nabla J_{S_{i-1}}(w)|.$$

在实验中有一个先验条件, 即在当前所做的假设条件下样本总体容量是不变的. 依据该规则得出以下公式, 且该公式用作后续公式代换:

$$J_S(w) = \sum_{i=1}^m \frac{1}{S_i} L(w; x_i), \quad (1)$$

其中 S_i 表示第 i 轮训练的样本容量, m 表示训练轮次数, S 表示均匀采样下的每一步训练的样本容量. 在原始的平均采样算法中, 目标函数为

$$J(w) = \frac{1}{S} \sum_{i=1}^m L(w; x_i),$$

其中函数 J 表示网络总体损失, S 表示每步训练的样本容量, $w = (w_1, w_2, \dots, w_n)$ 表示权重. 设原始算法的训练数据量的选择来源于 Fibonacci 采样算法, 在这种情况下, 构造函数 $\Psi_S(w)$ 表示这 2 种方法的损失差值, 得如下函数

$$\Psi_S(w) = \|J(w) - J_S(w)\| > \theta, \theta \in \mathbf{Z}^+.$$

对于给定的 $w \in \mathbf{R}^n$, $J_S(w)$ 对所有可能样本 $S \subset \{1, 2, \dots, N\}$ 的期望值等于 $J(w)$, 并且二者有梯度存在. 因此, Fibonacci 采样算法引入拉格朗日乘数 λ , 结合 (1) 式, 估计的样本数量 $P(w)$ 满足如下规则

$$P(w) = \|J(w) - J_S(w)\| - \sum_{i=1}^m \lambda_i \left(\sum_{i=1}^m S_i - mS \right),$$

由此可以得到 S, S_i 和 λ 的梯度如下:

$$\partial P(w) / \partial S = -\frac{1}{S^2} \sum_{i=1}^m L(w; x_i) + m \sum_{i=1}^m \lambda_i,$$

$$\partial P(w) / \partial S_i = \sum_{i=1}^m \frac{1}{S_i^2} L(w; x_i) - \sum_{i=1}^m \lambda_i,$$

$$\partial P(w) / \partial \lambda_i = \sum_{i=1}^m S_i - mS.$$

应用这些因子的导数性质,令 $\partial P(w) / \partial S = 0$, $\partial P(w) / \partial S_i = 0$, $\partial P(w) / \partial \lambda_i = 0$ 可得

$$\frac{1}{mS^2} \sum_{i=1}^m L(w; x_i) = \sum_{i=1}^m \frac{1}{S_i^2} L(w; x_i).$$

根据上述结论,由于同时对 2 个正数取平方不会改变 2 个数大小的关系,为便于后续推导 Fibonacci 采样算法使用函数 Ψ 近似,对 $J^2(w)$ 、 $J_s^2(w)$ 得出如下规则:

$$\Psi_s(w) = \|J^2(w) - J_s^2(w)\|.$$

对其进行扩展得

$$\Psi_s(w) = \left\| \frac{1}{S^2} \sum_{i=1}^m L(w; x_i) - \sum_{i=1}^m \frac{1}{S_i^2} L(w; x_i) \right\|.$$

由此得 $J(w)$ 和 $J_s(w)$ 的关系为

$$\Psi_s(w) = \left\| \frac{1}{S^2} \sum_{i=1}^m L(w; x_i) - \frac{1}{mS^2} \sum_{i=1}^m L(w; x_i) \right\|.$$

因此,得到 $\Psi_s(w) > \theta > 0$, 这表明 Fibonacci 采样算法相比于平均采样可以减少损失。

接下来描述在 Fibonacci 采样算法中最小化给定的凸目标函数 $J(w)$ 的算法。

先进行样本梯度算法初始化: 随机初始迭代权重 w_0 , 选择初始样本容量为 S_0, S_1 的数据样本并且设置常数 k 等于 0; 在超出范围的情况下设置最大阈值常数 T 。

输入: Fibonacci 批量图像数据;

输出: 分类精度以及分类损失;

重复如下步骤,直到数据实验完成:

(i) 计算梯度 $d_k = -\nabla J_{S_k}(w_k)$;

(ii) 定义新的迭代 $w_{k+1} = w_k + a_k d_k$, 其中 w_k 为第 k 轮的权重, a 为学习率;

(iii) 设置常数 $k \leftarrow k + 1$;

(iv) 从剩余样本中选择样本容量为 S_k 的批量投入模型训练, 并且 $|S_k| = |S_{k-1}| + |S_{k-2}|$;

(v) 若不满足 $S_k \leq T$ 则将 S_k 重新选择为 S_0 , 将

S_{k+1} 选择为 S_1 转到 (i)。

3 实验

3.1 SIGNS 数字手势数据集的分类

为了验证本文采样方法的效果, Fibonacci 采样算法进行了预测 SIGN 数字手势数据集上的数字类。SIGNS 数据集包含 1 080 个图像, 分别对应从数字 1~6 这 6 个数字手势。Fibonacci 采样算法使用了简单的卷积网络, 网络结构如图 1 所示。

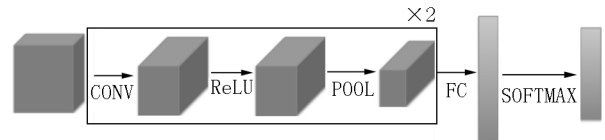


图 1 SIGNS 分类网络结构图

以 28×28 灰度图像作为输入。网络具有 2 个卷积层和 2 个池化层, 具有用于预测图像类的 softmax 层。在这种情况下, 卷积层使用 ReLU 作为非线性基准矫正方法, 在第 1 个卷积层和第 2 个卷积层之后分别使用 1 次最大池化层并且权重 w 被初始化为小的随机高斯值。第 2 个池化层之后是完全连接层, 具有 6 个输出单元, 最后一步引入交叉熵损失。Fibonacci 采样算法对网络进行了 100 个步骤的迭代, 在使用 Fibonacci 采样时, 每个小批量样本容量对应 1 个 Fibonacci 序列。本文对均匀批次的数量和 Fibonacci 数量采样对分类结果的影响比较感兴趣, 而不是在数字手势数据集上实现最先进的性能。

在表 1 中, baseline 是指采用原始均匀采样的结构相同的网络, 准确度是指测试准确度。表 1 结果验证了 Fibonacci 采样算法无论是在损失值上还是在预测准确度上均有较大的改善, 以批量数 16 为例, 原始均匀采样算法损失值为 0.690 397, Fibonacci 采样算法损失值为 0.263 681, 损失明显下降; 原始均匀采样算法预测准确度为 0.750 000, Fibonacci 采样算法预测准确度为 0.866 667。当批量数相等时, Fibonacci 采样策略可以在损失值或准确度上获得更好的结果。

Fibonacci 采样算法采用与最优化方法 Adam、

表 1 梯度下降的结果分析

| 批量数 | 损失(baseline 小批量) | 损失(Fibonacci 小批量) | 准确度(baseline) | 准确度(Fibonacci) |
|-----|-------------------|--------------------|----------------|-----------------|
| 16 | 0.690 397 | 0.263 681 | 0.750 000 | 0.866 667 |
| 32 | 1.367 832 | 0.482 913 | 0.441 667 | 0.875 000 |
| 48 | 1.664 303 | 0.806 149 | 0.341 667 | 0.708 333 |
| 64 | 1.735 111 | 1.006 628 | 0.300 000 | 0.658 333 |
| 128 | 1.770 775 | 1.498 073 | 0.266 667 | 0.491 667 |

注: Fibonacci 小批量大小是该处的最大阈值。

Momentum 和 RMSProp 进行比较,由于数据样本容量的选择, Fibonacci 采样算法采取的 Fibonacci 序列在训练相同轮次的前提下得到的损失值明显更小. 所以从另一角度来说, Fibonacci 采样算法得出的准确度相较于其他优化算法也会是更高的. 在各优化方法上的最佳准确度如表 2 所示, 实验结果表明 Fibonacci 采样算法的最佳准确度达到 95.80%.

表 2 最佳方法的比较

| 方法 | 最佳准确度 / % |
|----------------------|-----------|
| Minibatch + GD | 75.00 |
| Minibatch + Adam | 89.16 |
| Minibatch + Momentum | 85.00 |
| Minibatch + RMSProp | 87.50 |
| Fibonacci + GD | 95.80 |

3.2 cifar_10 数据集的分类

Fibonacci 采样算法将采用 AlexNet 原生网络结构, 并且将该方法应用于 cifar_10 数据集. 在 cifar_10 数据集中有 10 个类别的 60 000 张图片. Fibonacci 采样算法将数据集分为训练数据集和测试数据集 2 个部分, 使用随机的 50 000 张图片作为训练数据集, 10 000 张图片作为测试数据集. 实验结果皆在学习率为 0.000 01, 迭代次数是 1 000 等超参数设置下得出.

表 3 显示了 baseline 和 Fibonacci 小批量的不同优化方法的分类预测准确度. 从表 3 可以明确地看出 Fibonacci 小批量方法在 Adam、GD、Momentum 和 RMSProp 中有更好地改进, 以样本批量数 16 为例,

原始均匀采样方法下采用 Adam 优化方法达到的分类预测准确度为 0.553 2, 而在同等模型中 Fibonacci 采样方法下采用 Adam 优化方法达到的分类预测准确度为 0.592 3; 原始均匀采样方法下采用 Momentum 优化方法达到的分类预测准确度为 0.315 1, Fibonacci 采样方法下采用 Momentum 优化方法达到的分类预测准确度为 0.488 5; 原始均匀采样方法下采用 RMSProp 优化方法达到的分类预测准确度为 0.535 1, Fibonacci 采样方法下采用 RMSProp 优化方法达到的分类预测准确度为 0.549 6. 不同于类似的样本动态采样方法^[35], Fibonacci 采样算法在迭代中虽然有一点消耗, 但是并没有该方法计算下一批量大小的消耗.

然后进行耗时比较实验, 实验的设备是 Quadro P4000, 训练相同一批数据, 训练相同的完整轮次, 采用不同的采样方法和优化方法, 训练结果如表 4 所示. 以批量数 16 为例, 均匀采样结合 Adam 优化方法耗时约为 14 064.554 s, Fibonacci 采样结合 Adam 优化方法耗时约为 11 173.562 s; 均匀采样结合 Momentum 优化方法耗时约为 9 235.826 s, Fibonacci 采样结合 Momentum 优化方法耗时约为 13 451.496 s; 均匀采样结合 RMSProp 优化方法耗时约为 12 311.386 s, Fibonacci 采样结合 RMSProp 优化方法耗时约为 10 558.211 s. 结合表 3 的分类预测准确度分析, 在总体情况下, Fibonacci 采样方法相较于均匀采样方法是更优的.

表 3 优化结果的分析

| 批量数 | Adam | | GD | | Momentum | | RMSProp | |
|-------|----------|------------------|----------|------------------|----------|------------------|----------|------------------|
| | baseline | Fibonacci 小批量 | baseline | Fibonacci 小批量 | baseline | Fibonacci 小批量 | baseline | Fibonacci 小批量 |
| 16 | 0.553 2 | 0.592 3 | 0.149 2 | 0.102 6 | 0.315 1 | 0.488 5 | 0.535 1 | 0.549 6 |
| 32 | 0.603 2 | 0.644 8 | 0.104 0 | 0.175 1 | 0.381 2 | 0.581 7 | 0.578 7 | 0.602 2 |
| 48 | 0.641 9 | 0.674 6 | 0.104 2 | 0.201 1 | 0.427 3 | 0.639 8 | 0.616 3 | 0.638 1 |
| 64 | 0.662 0 | 0.688 7 | 0.104 2 | 0.207 1 | 0.457 1 | 0.634 4 | 0.632 5 | 0.653 4 |
| 100 | 0.670 8 | 0.694 2 | 0.104 4 | 0.234 3 | 0.469 6 | 0.663 1 | 0.648 9 | 0.663 8 |
| 128 | 0.679 1 | 0.702 2 | 0.104 4 | 0.255 5 | 0.479 6 | 0.681 2 | 0.653 2 | 0.679 9 |
| 256 | 0.684 4 | 0.717 0 | 0.104 9 | 0.263 4 | 0.484 6 | 0.690 1 | 0.663 0 | 0.684 2 |
| 500 | 0.687 2 | 0.719 5 | 0.105 2 | 0.266 8 | 0.490 7 | 0.691 2 | 0.666 7 | 0.694 6 |
| 1 000 | 0.686 9 | 0.723 9 | 0.105 4 | 0.270 3 | 0.490 6 | 0.689 2 | 0.667 6 | 0.697 5 |

最后本文进行了一次 Fibonacci 采样方法结合不同网络之间的分类测试准确度对比, 包括在 LeNet、AlexNet、VGG-16、GoogLeNet、ResNet50 等网络上进行实验, 实验的参数设置为 $e_{poch} = 1\ 000$ 、 $o_{optimizer} = Adam$ 、 $l_{earning_rate} = 0.000\ 1$ 、 $b_{atch_size} = 100$, 结果如表 5 所示.

在表 5 中各网络模型所得到的结果均是在 b_{atch_size} 为 100 的条件下得出的, 不同之处在于, 结合 Fibonacci 采样的模型中设置的 b_{atch_size} 是采样的最大阈值, 在 Fibonacci 采样算法实验中设置的起始值为 3 或 5. 对表 5 的数据进行比较分析, 当采用 LeNet 网络模型时, 原始均匀采样分类预测准确度是

64.01% ,Fibonacci 采样方法下分类预测准确度为 64.92%; 当采用 AlexNet 网络模型时 ,原始均匀采样分类预测准确度是 69.42% ,在 Fibonacci 采样方法下分类预测准确度为 70.47%; 当采用 VGG-16 网络模型时 ,原始均匀采样分类预测准确度是 73.41% ,在 Fibonacci 采样方法下分类预测准确度为 71.20%; 当采用 GoogLeNet 网络模型时 ,原始均

匀采样分类预测准确度是 73.30% ,在 Fibonacci 采样方法下分类预测准确度为 75.53%; 当采用 Res-Net50 网络模型时 ,原始均匀采样分类预测准确度是 74.51% ,在 Fibonacci 采样方法下分类预测准确度为 75.96% . 综上所述 ,Fibonacci 采样算法是切实可行的采样方法 ,有助于解决动态采样消耗大及经验采样的样本差异大的问题.

表 4 优化时间的分析

| 批量数 | Adam | | Momentum | | RMSProp | |
|-------|------------|------------|------------|------------|------------|------------|
| | baseline | Fibonacci | baseline | Fibonacci | baseline | Fibonacci |
| | | 小批量 | | 小批量 | | 小批量 |
| 16 | 14 064.554 | 11 173.562 | 9 235.826 | 13 451.496 | 12 311.386 | 10 558.211 |
| 32 | 45 021.588 | 11 080.310 | 9 666.575 | 14 910.355 | 39 425.910 | 10 337.545 |
| 48 | 10 887.979 | 12 347.953 | 8 273.119 | 73 324.920 | 9 186.813 | 10 432.804 |
| 64 | 10 603.682 | 11 548.034 | 7 941.352 | 18 022.427 | 9 225.327 | 8 615.386 |
| 100 | 8 747.933 | 11 152.552 | 11 884.052 | 11 405.040 | 8 142.515 | 7 820.927 |
| 128 | 7 388.020 | 23 517.359 | 40 472.123 | 13 519.732 | 7 390.911 | 8 839.248 |
| 256 | 12 598.904 | 8 067.948 | 7 017.564 | 40 800.918 | 14 346.677 | 8 593.023 |
| 500 | 7 179.421 | 8 428.812 | 6 871.393 | 11 728.424 | 41 859.896 | 25 119.841 |
| 1 000 | 7 845.639 | 7 362.126 | 6 533.157 | 8 174.280 | 7 711.424 | 7 408.553 |

表 5 不同网络模型下测试准确度的比较 %

| 网络模型 | 原始均匀采样 | Fibonacci 采样 |
|-----------|--------|--------------|
| LeNet | 64.01 | 64.92 |
| AlexNet | 69.42 | 70.47 |
| VGG-16 | 73.41 | 71.20 |
| GoogLeNet | 73.30 | 75.53 |
| ResNet50 | 74.51 | 75.96 |

4 结论

本文先提出了一种新的 Fibonacci 序列采样策略 ,同时验证了该采样方法是有效的; 然后 ,与小批量均匀采样方法进行了不同网络模型之间的比较以及耗时分析的比较; 最后 ,实验结果表明 Fibonacci 采样算法相较均匀采样得到了改进 ,在 SIGNS 数字数据集中最佳准确度达到了 95.80% . 在 cifar_10 数据集中 ,Fibonacci 采样算法通过运用不同的优化方法比较以及不同的网络模型比较得到了更优的结果. 但是 ,Fibonacci 采样算法暂时没有给出如何精确选择最小阈值、最大阈值的策略. 接下来的目标是给出这些选取策略.

5 参考文献

[1] Zhang Tong. Solving large scale linear prediction problems using stochastic gradient descent algorithms [EB/OL]. [2020-08-11]. [https://dl.acm.org/doi/abs/10.1145/](https://dl.acm.org/doi/abs/10.1145/1015330.1015332)

1015330.1015332.
[2] Rakhlin A ,Shamir O ,Sridharan K. Making gradient descent optimal for strongly convex stochastic optimization [EB/OL]. [2020-08-11]. <https://arxiv.org/abs/1109.5647>.
[3] Shamir O ,Zhang Tong. Stochastic gradient descent for non-smooth optimization: convergence results and optimal averaging schemes [EB/OL]. [2020-08-11]. <http://adsabs.harvard.edu/abs/2012arXiv1212.1824S>.
[4] Duchi J ,Singer Y. Efficient online and batch learning using forward backward splitting [J]. Journal of Machine Learning Research 2009 ,10: 2899-2934.
[5] Luo Zhiquan ,Tseng P. On the convergence of the coordinate descent method for convex differentiable minimization [J]. Journal of Optimization Theory and Applications , 1992 ,72(1) : 7-35.
[6] Mangasarian O L ,Musicant D R. Successive overrelaxation for support vector machines [J]. IEEE Transactions on Neural Networks 1999 ,10(5) : 1032-1037.
[7] Hsieh C J ,Chang Kaiwei ,Lin C J ,et al. A dual coordinate descent method for large-scale linear SVM [EB/OL]. [2020-08-11]. <http://dl.acm.org/citation.cfm?id=1390208>.
[8] Shalev-Shwartz S ,Tewari A. Stochastic methods for l_1 -regularized loss minimization [J]. Journal of Machine Learning Research 2011 ,12: 1865-1892.
[9] Lacoste-Julien S ,Jaggi M ,Schmidt M ,et al. Stochastic block-coordinate frank-wolfe optimization for structural

- SVMs [EB/OL]. [2020-08-11]. <http://arxiv.org/pdf/1207.4747v1.pdf>.
- [10] Nesterov Y. Efficiency of coordinate descent methods on huge-scale optimization problems [J]. *SIAM Journal on Optimization* 2012 22(2): 341-362.
- [11] Shalev-Shwartz S, Zhang Tong. Proximal stochastic dual coordinate ascent [EB/OL]. [2020-08-11]. <https://arxiv.org/abs/1211.2717>.
- [12] Shalev-Shwartz S, Zhang Tong. Stochastic dual coordinate ascent methods for regularized loss minimization [J]. *Journal of Machine Learning Research*, 2013, 14: 567-599.
- [13] Zhao Peilin, Zhang Tong. Stochastic optimization with importance sampling for regularized loss minimization [EB/OL]. [2020-08-11]. <https://dl.acm.org/doi/10.5555/3045118.3045120>.
- [14] 张林刚, 严广乐, 路晓伟. 嵌套分割算法: 一种新的并行随机优化算法 [J]. *计算机应用研究* 2007 24(6): 79-81.
- [15] 米永强, 高岳林. 求解约束优化问题的改进粒子群优化算法 [J]. *江西师范大学学报: 自然科学版*, 2015, 39(1): 59-63.
- [16] 夏红卫, 文传军. 一般非线性约束优化问题的信赖域法 [J]. *江西师范大学学报: 自然科学版*, 2012, 36(3): 253-256.
- [17] Robbins H, Monro S. A stochastic approximation method [J]. *The Annals of Mathematical Statistics*, 1951 22(3): 400-407.
- [18] Shalev-Shwartz S, Zhang Tong. Accelerated mini-batch stochastic dual coordinate ascent [EB/OL]. [2020-08-11]. <https://dl.acm.org/doi/10.5555/2999611.2999654>.
- [19] Defazio A, Bach F, Lacoste-Julien S. SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives [EB/OL]. [2020-08-11]. <https://dl.acm.org/doi/10.5555/2968826.2969010>.
- [20] Li Mu, Zhang Tong, Chen Yuqiang, et al. Efficient mini-batch training for stochastic optimization [EB/OL]. [2020-08-11]. <https://dl.acm.org/doi/10.1145/2623330.2623612>.
- [21] Luo Zhiquan, Tseng P. On the convergence of the coordinate descent method for convex differentiable minimization [J]. *Journal of Optimization Theory and Applications*, 1992 72(1): 7-35.
- [22] Tseng P. Convergence of a block coordinate descent method for nondifferentiable minimization [J]. *Journal of Optimization Theory and Applications*, 2001, 109(3): 475-494.
- [23] Saha A, Tewari A. On the nonasymptotic convergence of cyclic coordinate descent methods [J]. *SIAM Journal on Optimization* 2013 23(1): 576-601.
- [24] Wright S J. Coordinate descent algorithms [J]. *Mathematical Programming* 2015 151(1): 3-34.
- [25] Gurbuzbalaban M, Ozdaglar A, Parrilo P A, et al. When cyclic coordinate descent outperforms randomized coordinate descent [EB/OL]. [2020-08-11]. <http://mert.lids.mit.edu/w-content/uploads/sites/10/2017/11/CCDvsRCD.pdf>.
- [26] Nutini J, Schmidt M, Laradji I H, et al. Coordinate descent converges faster with the Gauss-Southwell rule than random selection [EB/OL]. [2020-08-11]. <http://arxiv.org/abs/1506.00552>.
- [27] Zheng Qu, Richtárik P. Coordinate descent with arbitrary sampling I: algorithms and complexity [J]. *Optimization Methods and Software* 2016 31(5): 829-857.
- [28] Stich S U, Raj A, Jaggi M. Approximate steepest coordinate descent [C] // Precup D, Teh Y W. *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, Aug 6-11, 2017. Sydney: PMLR, 2017, 70: 3251-3259.
- [29] Nesterov Y. Efficiency of coordinate descent methods on huge-scale optimization problems [J]. *SIAM Journal on Optimization* 2012 22(2): 341-362.
- [30] Richtárik P, Takáč M. Distributed coordinate descent method for learning with big data [J]. *Journal of Machine Learning Research* 2016 17: 75.
- [31] Richtárik P, Takáč M. On optimal probabilities in stochastic coordinate descent methods [J]. *Optimization Letters* 2016 10(6): 1233-1243.
- [32] Richtárik P, Takáč M. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function [J]. *Mathematical Programming*, 2014, 144(1/2): 1-38.
- [33] Lin Qihang, Lu Zhaosong, Xiao Lin. An accelerated proximal coordinate gradient method [EB/OL]. [2020-08-11]. <https://dl.acm.org/doi/10.5555/2969033.2969168>.
- [34] Byrd R H, Chin G M, Nocedal J, et al. Sample size selection in optimization methods for machine learning [J]. *Mathematical Programming* 2012 134(1): 127-155.
- [35] 杨杰明, 闫欣, 曲朝阳, 等. 基于数据密度分布的欠采样方法研究 [J]. *计算机应用研究* 2016 33(10): 2997-3000.
- [36] Shapiro A, Homem-de-Mello T. On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs [J]. *SIAM Journal on Optimization*, 2000 11(1): 70-86.
- [37] Shapiro A, Wardi Y. Convergence analysis of stochastic algorithms [J]. *Mathematics of Operations Research*, 1996, 21(3): 615-628.

- [38] Kleywegt A J, Shapiro A, Homem-de-Mello T. The sample average approximation method for stochastic discrete optimization [J]. *SIAM Journal on Optimization*, 2001, 12(2): 479-502.
- [39] Shapiro A, Homem-de-Mello T. A simulation-based approach to two-stage stochastic programming with recourse [J]. *Mathematical Programming*, 1998, 81(3): 301-325.
- [40] Homem-de-Mello T. Variable-sample methods for stochastic optimization [J]. *ACM Transactions on Modeling and Computer Simulation*, 2003, 13(2): 108-133.
- [41] Bastin F, Cirillo C, Toint P L. An adaptive Monte Carlo algorithm for computing mixed logit estimators [J]. *Computational Management Sciences*, 2006, 3(1): 55-79.
- [42] Sutskever I, Martens J, Dahl G, et al. On the importance of initialization and momentum in deep learning [EB/OL]. [2020-08-11]. <https://dl.acm.org/doi/10.5555/3042817.3043064>.
- [43] Qian Ning. On the momentum term in gradient descent learning algorithms [J]. *Neural Networks*, 1999, 12(1): 145-151.
- [44] Kingma D P, Ba J L. Adam: a method for stochastic optimization [EB/OL]. [2020-08-11]. <https://pubmed.ncbi.nlm.nih.gov/12662723/>.

The Study on Data Preprocessing Method Based on Fibonacci Sampling

DENG Hong^{1,3}, LIU Zhichao², PENG Yingqiong^{1,3*}, SHU Qing¹, HE Huojiao¹

(1. Software College, Jiangxi Agricultural University, Nanchang Jiangxi 330045, China;

2. Computer and Information Engineering, Jiangxi Agricultural University, Nanchang Jiangxi 330045, China;

3. Key Laboratory of Agricultural Information Technology of Jiangxi Province, Jiangxi Agricultural University, Nanchang Jiangxi 330045, China)

Abstract: Improving the convergence speed and prediction accuracy of neural network (NN) has always been a challenging problem in the field of computer artificial intelligence. Although many researchers have studied using small batches of data to train neural networks to obtain better results, these methods are not enough flexible. Aiming at this problem, the Fibonacci sampling algorithm that is a new data preprocessing algorithm is proposed. A new training data sequence is drawn according to the Fibonacci sequence rules, which can not only recover the small batch data partition, but also provide more flexible batch size selection. Experiments show that before gradient descent, the Fibonacci data partition sequence can get good experimental results. In the same single CNN (five-layer convolutional neural network) and AlexNet, the Fibonacci sampling algorithm can obtain higher accuracy and lower loss than traditional mini-batch gradient descent algorithm, and it can be used in several general networks such as LeNet, AlexNet, VGG-16, GoogLeNet.

Key words: mini-batch; Fibonacci sampling; neural network

(责任编辑: 冉小晓)