最优化大作业

162020117 袁靖宇

2022/05/29

1 概述：

本次大作业我选择阅读的论文为第三篇论文，即《Accelerating Stochastic Gradient Descent using Predictive Variance Reduction》。这篇论文里介绍了一种改进的随机梯度下降法，这一方法可以减少在迭代过程中对梯度的储存并采用显式方差缩减，并且具有较快的收敛速度。

2 文章要处理的问题以及问题的重要性

2.1 文章要处理的问题

一般的梯度下降法的有效性已经在过去的机器学习的发展中得到了充分的证明，也是如今深度学习能够在人工智能领域获得广泛应用的一个重要保证。然而，标准的梯度下降法每一次迭代都需要求 n 个导数，使得标准的梯度下降法在收敛时的计算量过大，时间花费过高。为了使用收敛性较好的梯度下降法并改进其计算方法来减少该算法在迭代过程中的计算量，提高计算效率和迭代效率以及收敛速度，一个较为流行的改进方法为随机梯度下降法（SGD），该算法每次随机地选择样本中的一个例子来近似所有样本来调整梯度，使得计算的复杂度变为标准梯度下降法的 $\frac{1}{n}$，减小了计算的复杂度，加快了收敛速率，并且最后的收敛结果在我们可以接受的范围内。但是随机梯度下降法的收敛性收到其方差的影响，为了保证其收敛性，降低算法运行过程中产生的方差，需要降低其学习率 η，使得其收敛速率存在一个所谓的“上限”。而作者所做的工作便是试图降低方差对收敛性的影响，降低算法迭代过程中的计算量，加快其收敛速度。

2.2 该问题的重要性

虽然从表面上来看，作者在本文中随提出的随机方差缩减法（SVRG）仅仅只是在对已有方法进行改进来增加其收敛速度，但是在对这一问题的研究中不可避免地会涉及到对于随机梯度方法中地方差以及收敛性与方差

之间产生影响的关系，在这一问题上的探究和成果能够更好地反映方差与收敛性的关系，对二者之间的关系产生更深的理解，推动相关算法的开发与灵感，同时，相对直观的方差减少也可以适用于非凸优化问题，相关的思想方法也可以用于训练深度神经网络。

3 该问题中的难点、他人对于这些难点的解决方案以及各自的优缺点

3.1 该问题中的难点：

该问题本质上其实是对于基础的梯度下降法的改进，正如前文中所谈到的，在标准的梯度下降法中，有效性以及收敛性已经得到了很好的证明，但是由于在算法运行过程中的每一步都要求 n 个导数，使得算法的计算成本过高导致收敛速率变慢，针对这一问题较为流行的改进方法为随机梯度下降法（SGD），此方法通过在每次迭代中随机选择样本中的一个例子来近似所有样本的梯度来通过每次将每一步迭代的求导次数优化为原来的 $\frac{1}{n}$ 来改进收敛速率。常见的随机梯度下降法的形式为：

$$w^{(t)} = w^{(t-1)} - \eta_t g_t(w^{(t-1)}, \xi_t), \tag{1}$$

然而，随机梯度下降法带来了新的问题，由于在计算时候随机地选择一部分来进行求导，这样的随机性引入了方差——这是由于 $g_t(w^{(t-1)}, \xi_t)$ 在期望中等同于梯度 $\bigtriangledown P(w^{(t-1)}$ 但是由于每一个 $g_t(w^{(t-1)}, \xi_t)$ 都是不同的。特别是，如果 $g_t(w^{(t-1)}, \xi_t)$ 很大，那么我们会有一个相对较大的方差，从而减慢收敛速度。也就是说在实际应用时我们被迫要权衡 SGD 的每次迭代计算速度快，收敛速度慢和梯度下降法的每次迭代计算慢、收敛快。这样的被迫权衡显然不是我们在改进梯度下降法时候希望看到的结果，故我们需要想办法降低方差来改进随机梯度下降法。

3.2 其他人对于这些难点的解决方案

Le Roux 等人在 2012 年提出了随机平均梯度法（SAG），Shalev-Shwartz 和 Zhang 在 2012 年提出了 SDCA 算法，这些方法适用于训练逻辑回归或最小二乘回归等凸线性预测问题。其中，随机平均梯度法（SAG）作为在 SGD 基础上的改进，与 SGD 相同的是，随机平均梯度法每次也随机取一个样本计算梯度，但是不同的是，随机平均梯度法实用所有样本梯度的平均值来更新参数。而 SDCA 算法则是结合了逻辑回归和 SVM 算法的最佳

特性和功能。

3.3 这些方法各自的优缺点

随机平均梯度由于保留了随机梯度下降法的随机选取，故也具有计算快的特点，同时，通过以平均值的方式来更新 w，使得损失值目标函数不易反复不收敛。这两个方法共同的缺点是其都需要储存所有的梯度或者其对偶变量，虽然在训练简单的正则化线性预测问题（如最小二乘回归等问题）中时并不是一个需要考虑和解决的问题，但是这一特点使得这些方法不适用于更复杂的应用程序。

4 本文中所使用的解决这一难点的方法及其优点

4.1 本文中所适用的方法

针对上述的问题和难点，作者提出了随机方差减少梯度算法（SVRG）。在随机梯度下降法（SGD）中，为了保证收敛学习率 η_t 衰减到 0，从而导致了收敛速度的变慢。对于这样一个较小方差的需求是由于 SGD 通过一部分样本来近似完整的梯度的过程中导致了方差的引入。故作者提出了一个修复方法。每一次，我们保持估计的 w 为 $\tilde{w}$ 来接近最优的 w，此外，我们保持平均梯度：

$$\tilde{\mu} = \bigtriangledown P(\tilde{w}) = \frac{1}{n}\sum_{i=1}^{n} \bigtriangledown \psi_i(\tilde{w}) \tag{2}$$

并且其计算需要使用 $\tilde{w}$ 对数据进行一次传递。需要注意的是，$\bigtriangledown \psi_i(\tilde{w}) - \tilde{\mu}$ 的期望是 0，因此下面的更新规则是广义的 SGD 算法：随机从 {1,...,n} 中抽取 i_t：

$$w^{(t)} = w^{(t-1)} - \eta_t(\bigtriangledown \psi_i(w^{(t-1)}) - \bigtriangledown \psi_{i_t}(\tilde{w}) + \tilde{\mu}) \tag{3}$$

我们因此有

$$E[w^{(t)}|w^{(t-1)}] = w^{(t-1)} - \eta_t \bigtriangledown P(w^{(t-1)}) \tag{4}$$

也就是说，如果我们让随机变量 $\psi_i = i_t$ 并且 $g_t(w^{(t-1)}, \psi_t) = \bigtriangledown \psi_{i_t}(w^{(t-1)}) - \bigtriangledown \psi_{i_t}(\tilde{w}) + \tilde{\mu}$, 那么，（3）就是（1）的特例。

其算法伪代码为：

4.2 这一方法的优点

1. 这一方法不需要储存完整的梯度，故可以适用于一些较为复杂的问题。

2. 与 SGD 算法不同，SVRG 算法中的学习率 η_t 不需要衰减，使得 SVRG 可以使用一个更大的学习率来获得更快的收敛速率。

3.SVRG 算法也可以适用于光滑但不强凸的问题。

Algorithm 1: SVRG

Input: 迭代次数 m，学习率 η

初始化 $\tilde{w_0}$;
for *s=1,2...* **do**
  $\tilde{w} = \tilde{w_{s-1}}$;
  $\tilde{\mu} = \frac{1}{n}\sum_{i=1}^{n} \bigtriangledown\psi_i(\tilde{w})$;
  $w_0 = \tilde{w}$ **for** *t=1,2...m* **do**
    随机选择 $i_t \in \{1, ..., n\}$ 并且更新权重;
    $w_t = w_{t-1} - \eta(\bigtriangledown\psi_{i_t}(w_{t-1}) - \bigtriangledown\psi_{i_t}(\tilde{w}) + \tilde{\mu})$
  end
  option1: 令 $\tilde{w_s} = wm$;
  option2: 随机选择 $t \in \{0, ..., m-1\}$, 令 $\tilde{w_s} = w_t$;
end

5 实验效果和与其他方法的对比

作者在实验中将 SVRG 与 SGD 和 SDCA 进行比较，将其应用在线性预测（凸）和神经网络（非凸）问题上。在所有的图中，x 轴表示梯度计算次数除以 n 来衡量的计算成本。对于 SGD，是通过训练数据的次数。对于非凸情况（神经网络）下的 SVRG，它包含了在每次迭代时和计算平均梯度 $\tilde{\mu}$ 时候对 $\bigtriangledown\psi_i(\tilde{w})$ 的额外计算。

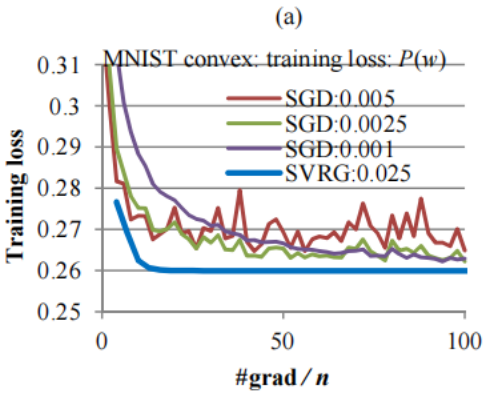
(a)
MNIST convex: training loss: P(w)
SGD:0.005
SGD:0.0025
SGD:0.001
SVRG:0.025
Training loss
0.31
0.3
0.29
0.28
0.27
0.26
0.25
0
50
100
#grad / n

图 1: 1

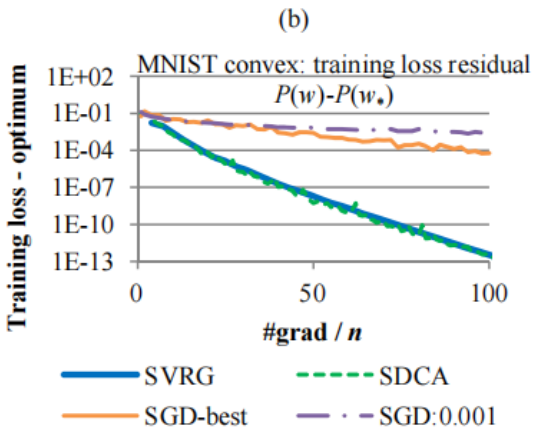
(b)
MNIST convex: training loss residual
P(w)-P(w*)
Training loss - optimum
1E+02
1E-01
1E-04
1E-07
1E-10
1E-13
0
50
100
#grad / n
SVRG
SDCA
SGD-best
SGD:0.001

图 2: 2

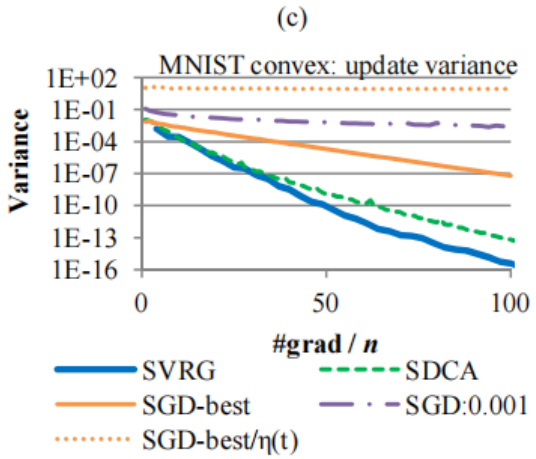
(c)
MNIST convex: update variance
1E+02
1E-01
1E-04
1E-07
1E-10
1E-13
1E-16
Variance
0
50
100
#grad / n
SVRG
SDCA
SGD-best
SGD:0.001
SGD-best/η(t)

图 3: 3