Sampled Quasi-Newton Methods for Deep Learning

Albert S. Berahas, Majid Jahani, Martin Takáč

Lehigh University, Bethlehem, PA

Abstract

✓ Proposed two novel quasi-Newton methods that use sampling to construct Hessian approximations
✓ Proved theoretical guarantees of the proposed methods
✓ Showed the practical performance of the methods on deep learning tasks
✓ Discussed the implementation details of the sampled quasi-Newton methods and compare them to the classical variants

Introduction

• and are of large, and F() is nonconvex
• First-order methods converge very slowly, and sometimes even fail to achieve 100% accuracy
• Methods that use the true Hessian are almost always able to achieve 100% in a few iterations; however, they are expensive
• The curvature information captured by classical quasi-Newton may not be adequate or useful
• Our idea: forget past curvature information and sample new curvature values at every iteration

Literature Review

• SR1/LSR1: Conn et al., 1991; Khatib et al., 1993; Byrd et al., 1996; Liu et al., 1996; Brot et al., 2017
• Stochastic QN: Schraudolph et al., 2007; Mohri & Ribeiro, 2015; Byrd et al., 2016; Berahas et al., 2016; Curtis et al., 2016

Quasi-Newton Methods

BFGS and LBFGS

\[ w_{k+1} = w_k - \alpha_k H_k F'(w_k) \]

where \( H_k = \nabla^2 F(w_k) - \rho_k \nabla F(w_k) \nabla F(w_k) \]

and the curvature pairs \((\alpha_k, \rho_k)\): \( w_{k+1} = w_k - \alpha_k \nabla F(w_k) \)

SR1 and LSR1

\[ F'(w_k) \leq F'(w_{k+1}) + \frac{1}{2} \alpha_k \nabla F(w_k) \nabla F(w_k) \alpha_k \]

SRI condition

\[ |F'(w_k) - F'(w_{k+1})| \leq \kappa_{tr} H_k H_k \]

Sampled Quasi-Newton Methods

Can one capture better curvature via sampling?

Sampled LBFGS & Sampled LSR1

Algorithm 2 Sampled LBFGS (SLBFGS)

1. for \( k = 0, 1, 2, \ldots \) do
2. Compute new \((\alpha_k, \rho_k)\) pairs via Algorithm 1
3. Compute \( \rho_k = \langle \nabla F(v_k), \nabla F(w_k) \rangle \)
4. Choose the smallest \( \alpha_k > 0 \)
5. Set \( w_{k+1} = w_k + \alpha_k \nabla F(w_k) \)
6. end for

Theorem 4. Suppose that Assumptions 1 and 2 hold, and let \( F' = F'(w^*) \) where \( w^* \) is the minimizer of \( F \). Let \( (w_k) \) be the iterates generated by Algorithm 2, where \( 0 < \alpha_k \leq \alpha_{max} \) and \( w_k \) is the starting point. Then for all \( k \geq 0 \),

SLR1: Sampled LSR1 - Nonconvex Functions

Assumption 5. The function \( F(w) \) is bounded below by a scalar \( \hat{F} \).

Assumption 6. The gradients of \( F \) are L-Lipschitz continuous for all \( w \in \mathbb{R}^d \).

Theorem 8. Suppose that Assumptions 1 and 2 hold. Let \( (\bar{x}_k) \) be the iterates generated by Algorithm 2 with the modification that the inverse approximation update is performed using only curvature pairs that satisfy \( B \). Then, for some \( s > 0 \), and \( H_k = I \) if no curvature pairs satisfy \( B \). Then, there exist constants \( 0 < \mu_k \leq 2 \) such that

Sampled LSR1

Assumption 9. For all \( k \),

\[ m_k(0) - m_k(\bar{x}_k) \geq \frac{1}{2} \nabla F'(\bar{x}_k)^T \]

\[ \min \left[ \frac{\nabla F'(\bar{x}_k)^T \Delta_k}{\text{opt} \tau} \right] \]

Theorem 10. Suppose that Assumptions 1 and 2 hold. Let \( (\bar{x}_k) \) be the Hessian approximations generated by Algorithm 2 with the modification that the approximation update is performed using only curvature pairs that satisfy \( B \), for some \( s > 0 \), and \( H_k = I \) if no curvature pairs satisfy \( B \). Then, there exists a constant \( \mu_k > 0 \) such that

\[ \|H_k\| \leq \mu_k \] for \( k = 0, 1, 2, \ldots \)

Comparison of Computational Cost and Storage

• Numbers of line search iterations and CG iterations are denoted as \( n_{ns} \) and \( n_{cg} \), respectively

• The per iteration cost of the sampled quasi-Newton methods is COMPARABLE to that of the classical limited memory variants

• In \( m \times n \) regime, the computational cost of the methods is \( O(n^2) \)

Numerical Results

Toy Classification Problem

• Pure classes each with 50 data points
• Trained these FCN models in COMPARABLE time as the classical limited memory variants

Performance metrics:

- Images / second
- Model
- VGG-A
- LeNet
- Alexnet v2

- Method
- Function Value
- Hessian Vector
- Gradient

Distributed Computing

Performance (Images/second) as a function of batch size for different DNN models and operations on a single P100 GPU (left). Time (seconds) to complete 1 epoch of SGD and to perform 1 iteration of SLSR1 on a dataset with 1M images using varying number of MPS processes.

Future Work

- Extend our proposed methods to the stochastic setting (inexact gradients and/or Hessians)
- Extend and modify our methods to incorporate adaptive batch sizes and memory
- Conduct a large scale numerical investigation of the proposed methods

References


- Code: https://github.com/CS284斯坦福大学