Stochastic Quasi Newton Method

Parth L   Radhika G   Sriram K   Vinod L

Department of Computer Science
Stony Brook

May 01, 2018

Outline

1. Motivation
2. Recap
3. Stochastic Quasi Newton implementation
   - Assumption
   - Algorithm
   - Contribution
   - Issues faced
4. Results
5. Summary
6. References

Motivation

- LBFGS is an unique algorithm as it has a quadratic update with the computational complexity per iteration $O(d^2)$
- Stochastic update for non convex functions is needed as most of the deep learning applications are nonconvex
- Pytorch is widely used in most of deep learning applications
- State of Art
    - SGD iteration complexity is $O(\epsilon^{-2})$ and it is the best we can achieve in any stochastic.

Recap

- Newton method is super linear in convergence It is expensive as it requires second order derivatives.

$$x_{k+1} = x_k - t\nabla^2 f(x_k)^{-1}\nabla f(x_k) \quad (1)$$

- We can use the approximation for Hessian inverse in quasi newton methods i.e

$$H_k^{-1} = (I - \frac{sy^T}{y^Ts})H_{k-1}^{-1}(I - \frac{ys^T}{y^Ts}) + \frac{ss^T}{y^Ts} \quad (2)$$

where

$$s = x^{(k)} - x^{(k-1)}, y = \nabla f(x^k) - \nabla f(x^{(k-1)}) \quad (3)$$

Stochastic Optimization Problem

- We consider the following stochastic optimization problem, i.e

$$\min f(x) = E[F(x, \xi)] \tag{4}$$

where $F : R^n x R^d \rightarrow R$ is continuously differentiable and possibly nonconvex, $\xi : R^d$ denotes a random variable with distribution function P

Outline

1. Motivation
2. Recap
3. Stochastic Quasi Newton implementation
   - Assumption
   - Algorithm
   - Contribution
   - Issues faced
4. Results
5. Summary
6. References

Assumptions

- f is continuously differentiable and ∇f is globally Lipschitz continuous with L i.e $||\nabla f(x) - \nabla f(y)|| \leq L||x - y||$
- For any iteration k, we have $E_{\xi_k}[g(x_k, \xi_k] = \nabla f(x_k)$ and $E_{\xi_k}[||g(x_k, \xi_k) - \nabla f(x_k)||^2] \leq \sigma^2$
- There exists two positive constants m M such that $mI \succeq H_k \succeq MI$ where the notation $A \succeq B$ with A,B belongs to R^{nxn}
- For any $k \geq 2$, $E[H_k g_k|\xi_{k-1}] = H_k * \nabla f(x_k)$

Outline

1. Motivation
2. Recap
3. Stochastic Quasi Newton implementation
   - Assumption
   - Algorithm
   - Contribution
   - Issues faced
4. Results
5. Summary
6. References

Positive definite condition for non convex functions

- Algorithm

$$\bar{y}_{k-1} = \hat{\theta}_{k-1} y_{k-1} + (1 - \hat{\theta}_{k-1}) B_{k-1} s_{k-1},$$

$$\hat{\theta}_{k-1} = \begin{cases} \frac{0.75 s_{k-1}^\top B_{k-1} s_{k-1}}{s_{k-1}^\top B_{k-1} s_{k-1} - s_{k-1}^\top y_{k-1}} & \text{if } s_{k-1}^\top y_{k-1} < 0.25 s_{k-1}^\top B_{k-1} s_{k-1}, \\ 1 & \text{otherwise.} \end{cases}$$

$$y_{k-1} := \bar{g}_k - g_{k-1} = \frac{\sum_{i=1}^{m_{k-1}} g(x_k, \xi_{k-1,i}) - g(x_{k-1}, \xi_{k-1,i})}{m_{k-1}}.$$

where this update ensures the positive definiteness of H_k

Algorithm

$$\rho_{k-1} = (s_{k-1}^\top \bar{y}_{k-1})^{-1}$$

$$H_{k,0} = \gamma_k^{-1} I, \quad \text{where } \gamma_k = \max\left\{ \frac{y_{k-1}^\top y_{k-1}}{s_{k-1}^\top y_{k-1}}, \delta \right\} \geq \delta,$$

Input: Let x_k be a current iterate. Given the stochastic gradient g_{k-1} at iterate x_{k-1}, the random variable ξ_{k-1}, the batch size m_{k-1}, s_j, $\bar{y}_j$ and ρ_j, $j = k-p, \ldots, k-2$, and $u_0 = g_k$.

Output: $H_k g_k = v_p$.

Set $s_{k-1} = x_k - x_{k-1}$ and calculate y_{k-1}
Calculate γ_k
Calculate $\bar{y}_{k-1}$
for $i = 0, \ldots, \min\{p, k-1\} - 1$ **do**
    Calculate $\mu_i = \rho_{k-i-1} u_i^\top s_{k-i-1}$
    Calculate $u_{i+1} = u_i - \mu_i \bar{y}_{k-i-1}$
end for
Calculate $v_0 = \gamma_k^{-1} u_p$
for $i = 0, \ldots, \min\{p, k-1\} - 1$ **do**
    Calculate $\nu_i = \rho_{k-p+i} v_i^\top \bar{y}_{k-p+i}$
    Calculate $v_{i+1} = v_i + (\mu_{p-i-1} - \nu_i) s_{k-p+i}$.
end for

Outline

1. Motivation
2. Recap
3. Stochastic Quasi Newton implementation
   - Assumption
   - Algorithm
   - Contribution
   - Issues faced
4. Results
5. Summary
6. References

Contribution

- Update of Hessian from scalar quantity to d*d
- Updation of Hessian for non convex function such that H_k remains positive definite
- Stochastic minibatch update for the y_k

Outline

1. Motivation
2. Recap
3. Stochastic Quasi Newton implementation
   - Assumption
   - Algorithm
   - Contribution
   - Issues faced
4. Results
5. Summary
6. References

Issues faced

- Hessian is a scalar in the existing LBFGS code
- Stochastic mini batch update for the algorithm
- Incorrect test cases present in pytorch
- Lack of support and documentation of LBFGS code on pytorch
- Unavailability of GPUs, which slowed down experimentation
- Lack of existing deep learning experiments with Quasi-newton techniques

Results

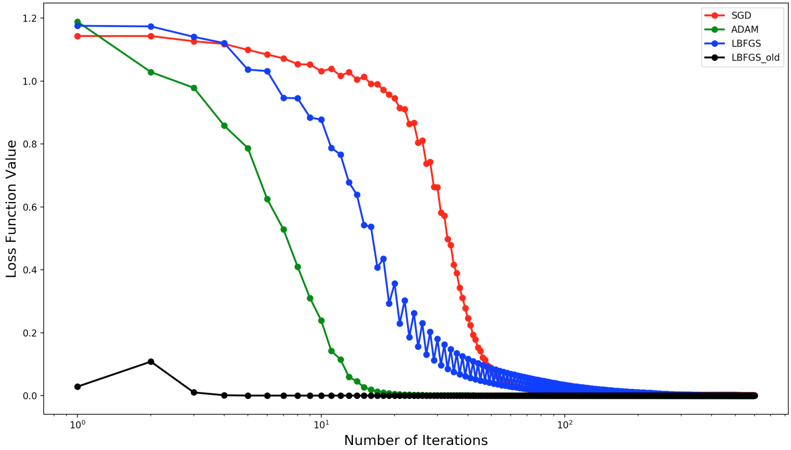

Figure: Comparison of change in loss function of SGD, ADAM, LBFGS, and SdLBFGS over an LSTM network

Results

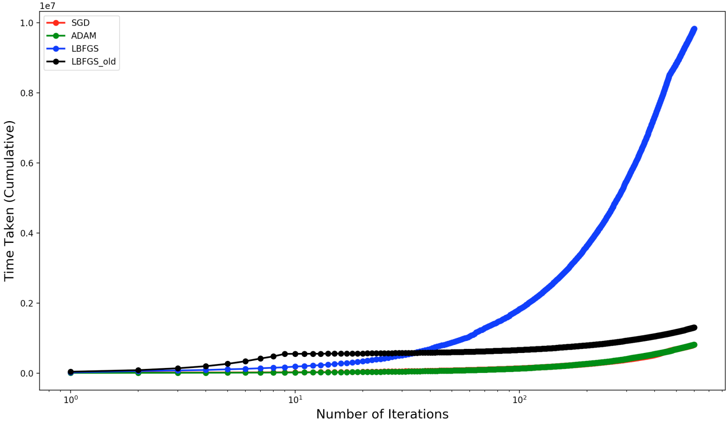

Figure: The total time taken for the four optimization algorithms. SdLBFGS takes the longest time

Results

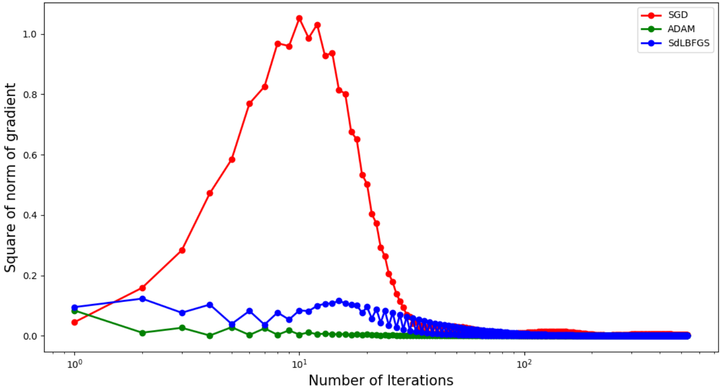

Figure: Change in square of norm of gradient over time

Results

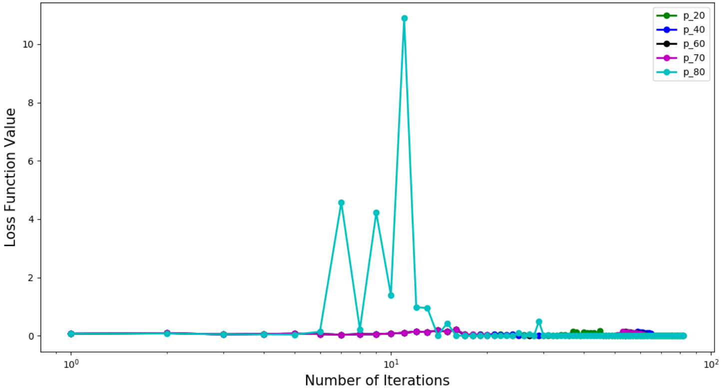

Figure: Change in the loss value with a change in the history size denoted by p

Summary

- SDLBFGS gives better ϵ suboptimality than SGD but the over all time is more for SDLBFGS
- Adam gives best performance when compared to SGD SDLBFGS

References

1. Wang, Xiao, et al. "Stochastic quasi-Newton methods for nonconvex stochastic optimization." SIAM Journal on Optimization 27.2 (2017): 927-956.
2. http://aria42.com/blog/2014/12/understanding-lbfgs
3. http://pytorch.org/tutorials/
4. https://discuss.pytorch.org/
5. http://ruder.io/open