A Stochastic Momentum Accelerated Quasi-Newton Method for Neural Networks (Student Abstract)

S. Indrapriyadarsini [1], Shahrzad Mahboubi [2], Hiroshi Ninomiya [2], Takeshi Kamio [3], Hideki Asai [4]

[1] Graduate School of Science and Technology, Shizuoka University, Japan
[2] Graduate School of Electrical and Information Engineering, Shonan Institute of Technology, Japan
[3] Graduate School of Information Sciences, Hiroshima City University, Japan
[4] Research Institute of Electronics, Shizuoka University, Japan
{s.indrapriyadarsini.17, asai.hideki}@shizuoka.ac.jp, {20T2502@sit, ninomiya@info}.shonan-it.ac.jp,
kamio@hiroshima-cu.ac.jp

Abstract

Incorporating curvature information in stochastic methods has been a challenging task. This paper proposes a momentum accelerated BFGS quasi-Newton method in both its full and limited memory forms, for solving stochastic large scale non-convex optimization problems in neural networks (NN).

Introduction

A majority of recent applications employ large NN models trained using massive amounts of data, thereby imposing high computational load and storage memory. Hence, there is a great demand for large scale stochastic algorithms that can train NNs based on a relatively small subset of the training data. Gradient based methods are popularly used in training NNs and can be broadly classified as first and second order methods. Despite the high computational cost, second order methods such as the BFGS quasi-Newton method have shown to have faster convergence compared to first order methods. Incorporating second order curvature information in stochastic settings is a challenging task and has been an active area of research. This paper proposes a stochastic (online) momentum accelerated quasi-Newton method in both its full and limited memory forms for solving large scale non-convex optimization problems in neural networks.

Background

Simple first order stochastic gradient descent takes the form

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha_k \nabla \mathbf{E}(\mathbf{w}_k, \mathbf{X}_k), \tag{1}$$

where $\nabla \mathbf{E}(\mathbf{w}_k, \mathbf{X}_k)$ is the gradient of the error function computed on a mini-batch $\mathbf{X} \subseteq T_r$, and α_k is the learning rate, usually determined by a decay schedule. The oBFGS method (Schraudolph et al. 2007) is one of the early scalable and stable stochastic quasi-Newton methods and is given as

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha_k \mathbf{H}_k^{\mathrm{BFGS}} \nabla \mathbf{E}(\mathbf{w}_k, \mathbf{X}_k), \tag{2}$$

where $\mathbf{H}_k^{\mathrm{BFGS}}$ is a symmetric positive definite matrix iteratively approximated by the following BFGS formula.

$$\mathbf{H}_{k+1}^{\mathrm{BFGS}} = (\mathbf{I} - \rho_k \mathbf{p}_k \mathbf{q}_k^{\mathrm{T}}) \mathbf{H}_k^{\mathrm{BFGS}} (\mathbf{I} - \rho_k \mathbf{q}_k \mathbf{p}_k^{\mathrm{T}}) + \rho_k \mathbf{p}_k \mathbf{p}_k^{\mathrm{T}}, \tag{3}$$

$$\rho_k = 1/\mathbf{q}_k^{\mathrm{T}} \mathbf{p}_k, \quad \mathbf{p}_k = \mathbf{w}_{k+1} - \mathbf{w}_k,$$
$$\text{and } \mathbf{q}_k = \nabla \mathbf{E}(\mathbf{w}_{k+1}, \mathbf{X}_k) - \nabla \mathbf{E}(\mathbf{w}_k, \mathbf{X}_k). \tag{4}$$

In the limited memory form, the search direction $\mathbf{g}_k = \mathbf{H}_k^{\mathrm{BFGS}} \nabla \mathbf{E}(\mathbf{w}_k, \mathbf{X}_k)$ is determined by the two loop recursion (Nocedal and Wright 2006). Note that in (4), o(L)BFGS computes the gradient twice using the same mini-batch sample to reduce sampling noise and ensure scalability.

Similar to the o(L)BFGS, the stochastic variant of NAQ (Ninomiya 2017), namely, o(L)NAQ (Indrapriyadarsini et al. 2019) was shown to accelerate o(L)BFGS. o(L)NAQ also computes two gradients per iteration, one of which is the Nesterov's accelerated gradient $\nabla \mathbf{E}(\mathbf{w}_k + \mu \mathbf{v}_k, \mathbf{X}_k)$. The momentum parameter is chosen as $0 < \mu < 1$. The update equations for NAQ are as follows.

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \mathbf{v}_{k+1}, \tag{5}$$

$$\mathbf{v}_{k+1} = \mu \mathbf{v}_k - \alpha_k \mathbf{H}_k^{\mathrm{NAQ}} \nabla \mathbf{E}(\mathbf{w}_k + \mu \mathbf{v}_k, \mathbf{X}_k), \tag{6}$$

where $\mathbf{H}_k^{\mathrm{NAQ}}$ is symmetric positive definite matrix given by

$$\mathbf{H}_{k+1}^{\mathrm{NAQ}} = (\mathbf{I} - \rho_k \hat{\mathbf{p}}_k \hat{\mathbf{q}}_k^{\mathrm{T}}) \mathbf{H}_k^{\mathrm{NAQ}} (\mathbf{I} - \rho_k \hat{\mathbf{q}}_k \hat{\mathbf{p}}_k^{\mathrm{T}}) + \rho_k \hat{\mathbf{p}}_k \hat{\mathbf{p}}_k^{\mathrm{T}}, \tag{7}$$

$$\rho_k = 1/\hat{\mathbf{q}}_k^{\mathrm{T}} \hat{\mathbf{p}}_k, \quad \hat{\mathbf{p}}_k = \mathbf{w}_{k+1} - (\mathbf{w}_k + \mu \mathbf{v}_k),$$
$$\text{and } \hat{\mathbf{q}}_k = \nabla \mathbf{E}(\mathbf{w}_{k+1}, \mathbf{X}_k) - \nabla \mathbf{E}(\mathbf{w}_k + \mu \mathbf{v}_k, \mathbf{X}_k). \tag{8}$$

The limited memory oLNAQ is formulated by computing the search direction $\mathbf{g}_k = \mathbf{H}_k^{\mathrm{NAQ}} \nabla \mathbf{E}(\mathbf{w}_k + \mu \mathbf{v}_k, \mathbf{X}_k)$ using the two-loop recursion.

Proposed Algorithm : o(L)MoQ

The Momentum Quasi-Newton (MoQ) method (Mahboubi et al. 2019) showed that the Nesterov's accelerated gradient in NAQ can be approximated as a linear combination of past gradients as shown below.

$$\nabla \mathbf{E}(\mathbf{w}_k + \mu \mathbf{v}_k) \approx (1 + \mu) \nabla \mathbf{E}(\mathbf{w}_k) - \mu \nabla \mathbf{E}(\mathbf{w}_{k-1}). \tag{9}$$

Extending this approximation to o(L)NAQ, this paper proposes a stochastic momentum accelerated quasi-Newton (oMoQ) method. The algorithm is as shown in Algorithm 1. The Hessian $\mathbf{H}_k^{\mathrm{MoQ}}$ is updated by (10) where $\mathbf{s}_k$ and $\mathbf{y}_k$ are computed as shown in steps 12 and 13 of Algorithm 1.

$$\mathbf{H}_{k+1}^{\mathrm{MoQ}} = (\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}_k^{\mathrm{T}}) \mathbf{H}_k^{\mathrm{MoQ}} (\mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^{\mathrm{T}}) + \rho_k \mathbf{s}_k \mathbf{s}_k^{\mathrm{T}}. \tag{10}$$

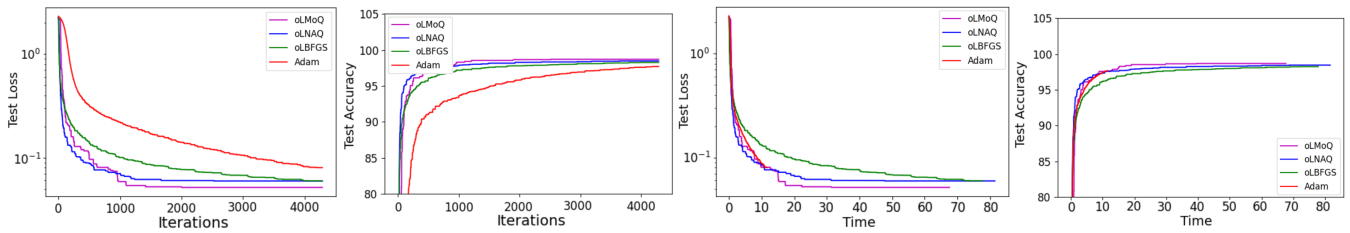

Figure 1: LeNet-5 results on 28x28 MNIST with $b = 128, m = 4$

Note that unlike o(L)BFGS and o(L)NAQ, the proposed method computes only one gradient per iteration, while the gradient computed with respect to the previous batch $\mathbf{X}_k$ is stored in memory. The computation cost is thus reduced by bd, where b is the batch size and d is the number of parameters. Limited memory oLMoQ is formulated by computing the search direction $\mathbf{g}_k$ (step 5) using the two-loop recursion.

Simulation Results

The performance of oLMoQ is evaluated on 28x28 MNIST dataset and LeNet-5 architecture with a batch size $b = 128$, limited memory $m = 4$, $\mu = 0.85$, $d = 61,706$ for 10 epochs. Based on experimentation, a decay schedule of $\alpha_{t+1} = 0.5\alpha_t$ is chosen for oLMoQ, where t is the epoch. The α_k schedules of oLBFGS and oLNAQ are chosen as in (Schraudolph et al. 2007) and (Indrapriyadarsini et al. 2019), respectively. From the iteration versus test loss and test accuracy graphs, we can observe that the momentum methods oLNAQ and oLMoQ perform better than Adam and oLBFGS, and oLMoQ performs better than oLNAQ. From the time versus test loss and test accuracy graphs, we can see that oLMoQ is faster compared to oLBFGS and oLNAQ, which is due to one gradient computation per iteration. Though the per iteration time of Adam is still better than oLMoQ, we can notice that Adam would take a few more iterations to reach the same test loss or test accuracy value as that of oLMoQ. Finally, Table 1 shows the summary of computation and storage costs.

Algorithm	Computational Cost	Storage
oBFGS	$2bd + d^2$	d^2
oLBFGS	$2bd + 6md$	$2md$
oNAQ	$2bd + d^2$	d^2
oLNAQ	$2bd + 6md$	$2md$
oMoQ	$bd + d^2$	$d^2 + d$
oLMoQ	$bd + 6md$	$(2m + 1)d$

Table 1: Summary of Computional Cost.

Algorithm 1: Stochastic MoQ

Require: learning rate schedule, $0 < \mu < 1$ and k_{max}
Ensure: $\mathbf{w}_k \in \mathbb{R}^d$, $\mathbf{H}_k = \epsilon\mathbf{I}$ and $\mathbf{v}_k = 0$
1: Calculate $\nabla\mathbf{E}(\mathbf{w}_k, X_k)$
2: **while** $||\nabla\mathbf{E}(\mathbf{w}_k)|| > \epsilon$ and $k < k_{\max}$ **do**
3: Determine learning rate α_k
4: $\nabla\mathbf{E}_1 = (1+\mu)\nabla\mathbf{E}(\mathbf{w}_k, \mathbf{X}_k) - \mu\nabla\mathbf{E}(\mathbf{w}_{k-1}, \mathbf{X}_{k-1})$
5: $\mathbf{g}_k \leftarrow -\mathbf{H}_k\nabla\mathbf{E}_1$
6: $\mathbf{g}_k = \mathbf{g}_k/||\mathbf{g}_k||_2$
7: $\mathbf{v}_{k+1} \leftarrow \mu\mathbf{v}_k + \alpha_k\mathbf{g}_k$
8: $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + \mathbf{v}_{k+1}$
9: Store $\nabla\mathbf{E}(\mathbf{w}_k, \mathbf{X}_k)$
10: Select mini-batch $\mathbf{X}_{k+1}$
11: Calculate $\nabla\mathbf{E}_2 = \nabla\mathbf{E}(\mathbf{w}_{k+1}, \mathbf{X}_{k+1})$
12: $\mathbf{s}_k \leftarrow \mathbf{w}_{k+1} - (\mathbf{w}_k + \mu\mathbf{v}_k)$
13: $\mathbf{y}_k \leftarrow \nabla\mathbf{E}_2 - \nabla\mathbf{E}_1 + \lambda\mathbf{s}_k$
14: Update $\mathbf{H}_k$ using (10)
15: **end while**

Conclusion

This paper proposed a stochastic momentum accelerated quasi-Newton method, and evaluated its performance in comparison to oLBFGS and oLNAQ on the 28×28 MNIST classification problem. The proposed oLMoQ method computes only one gradient per iteration and has shown to converge faster compared to oLBFGS and oLMoQ. In full batch setting, NAQ (Ninomiya 2017) and MoQ (Mahboubi et al. 2019) have shown to have convergence properties similar to that of the BFGS method. As future works, the convergence properties of these methods in the stochastic setting will be studied along with the effectiveness on larger problems.

References

Indrapriyadarsini, S.; Mahboubi, S.; Ninomiya, H.; and Asai, H. 2019. A Stochastic Quasi-Newton Method with Nesterov's Accelerated Gradient. In *ECML-PKDD*. Springer.

Mahboubi, S.; Indrapriyadarsini, S.; Ninomiya, H.; and Asai, H. 2019. Momentum Acceleration of Quasi-Newton Training for Neural Networks. In *Pacific Rim International Conference on Artificial Intelligence*, 268–281. Springer.

Ninomiya, H. 2017. A novel quasi-Newton-based optimization for neural network training incorporating Nesterov's accelerated gradient. *Nonlinear Theory and Its Applications, IEICE*, 8(4): 289–301.

Nocedal, J.; and Wright, S. J. 2006. *Numerical Optimization*. Springer Series in Operations Research. Springer, second edition.

Schraudolph, N. N.; Yu, J.; Günter, S.; and . 2007. A stochastic quasi-Newton method for online convex optimization. In *Artificial Intelligence and Statistics*, 436–443.