OPT2020: 12th Annual Workshop on Optimization for Machine Learning

Learning To Combine Quasi-Newton Methods

Maojia Li MXL8487@RIT.EDU
Rochester, NY, USA

Jialin Liu JIALIN.LIU@ALIBABA-INC.COM
Bellevue, WA, USA

Wotao Yin WOTAOYIN@MATH.UCLA.EDU
Los Angeles, CA, USA

Abstract

Applying machine learning techniques to accelerate optimization algorithms has recently gained researchers' attention. Different from manually-designed update rules, learning-based algorithms "learn to optimize" from data. When solving different but similar optimization problems repetitively, learning-based algorithms update their iterates using previous problem-solving experiences. This paper describes our study on a learning-based Quasi-Newton framework. Our study uses two recurrent neural networks that, at each iteration, select a step size and a Hessian approximator from a modified Broyden class. We train them offline to accelerate online convergence. The proposed framework shows outstanding performance on logistic regression. In the future, we will extend our study to additional classes of optimization problems.

1. Introduction

Quasi-Newton methods such as the Davidon-Fletcher-Powell (DFP) method [11, 14], Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [4, 13, 15, 25], and Symmetric Rank 1 (SR1) method [11] replace the Hessian in Newton's method by computationally-cheaper approximations. They are among the most efficient methods for smooth unconstrained optimization problems. In [5, 21], it was shown that under certain assumptions, quasi-Newton methods equipped with a practical line search could achieve a superlinear convergence rate. The efficiency of quasi-newton methods is often related to its speed of recovering true Hessian. A quasi-newton method is often equipped with an exact or inexact line search that ensures the curvature condition.

Motivated by the recent work [23], which introduces an optimization scheme that selects different Hessian approximators at each iteration from a large subclass of Quasi-Newton methods, we propose a data-driven method that adaptively selects an approximator for the inverse Hessian (unlike [23]), as well as a step size, to achieve a faster practical speed. Specifically, a learning-based framework will select a Hessian inverse approximator from a sub-class of the Broyden family.

Consider $f : \mathbb{R}^n \to \mathbb{R}$, which is at least twice continuously differentiable. A quasi-Newton method generates a new iterate:

$$x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k),$$

where α_k is the step size and H_k is the inverse Hessian approximate at k^{th} iterate x_k. BFGS and DFP inverse Hessian approximates are, respectively,

$$H_{k+1}^{\text{BFGS}} = H_k - \frac{s_k y_k^T H_k + H_k y_k s_k^T}{y_k^T s_k} + \left(\frac{y_k^T H_k y_k}{y_k^T s_k} + 1\right)\frac{s_k s_k^T}{y_k^T s_k}$$

and

$$H_{k+1}^{\text{DFP}} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k},$$

for

$$s_k := x_{k+1} - x_k, \quad y_k := \nabla f(x_{k+1}) - \nabla f(x_k).$$

Under $H_k \succ 0$ and curvature condition $y_k^T s_k > 0$, H_{k+1}^{BFGS} and H_{k+1}^{DFP} are also positive definite.

The Broyden class is a family of quasi-Newton methods satisfying secant condition $H_{k+1} y_k = s_k$ [6, 12, 20, 22] defined as

$$H_{k+1}^{\text{BFGS-DFP}}(\tau) = \tau H_{k+1}^{\text{BFGS}} + (1-\tau) H_{k+1}^{\text{DFP}}, \tag{1}$$

where $\tau \in \mathbb{R}$ is the Broyden parameter. The restricted Broyden class has $\tau \in [0, 1]$. SR1

$$H_{k+1}^{\text{SR1}} = H_k - \frac{(H_k y_k - s_k)(H_k y_k - s_k)^T}{(H_k y_k - s_k)^T y_k}. \tag{2}$$

corresponds (1) by setting $\tau = \frac{y_k^T s_k}{(s_k - H_k y_k)^T y_k}$. While H_{k+1}^{SR1} exhibits faster convergence than BFGS on many problems, H_{k+1}^{SR1} is not necessarily positive definite and the denominator of (2) can vanish. There exist convex quadratic problems [10] on which no SR1 update satisfies the secant condition.

Our method finds a sequence of high-quality parameters (τ_k, α_k) using meta-learning. Rather than switching between well-known approximators in [1, 2, 13], we introduce an alternative parameterization of the Broyden class and use training data to obtain a model that selects $\tau_k \in [0, 1]$ adaptively. The model consists of two recurrent neural networks (RNNs) [24] that find the parameter pair at each iteration k based on information accumulated from the previous $k-1$ steps. The RNN used in this paper is long short-term memory (LSTM) [17].

Our approach belongs to the class of methods known as *learning to optimize* (L2O). In L2O, neural networks are trained to obtain efficient models for a particular distribution of data. This results in task-specific algorithms [3, 7, 8, 19] that can converge order(s) of magnitude faster than general-purpose counterparts on problems sufficiently similar to those used in training. Based on our own experience and private communication, it is generally difficult to obtain excellent performance on L2O methods using *second-order information*. However, we present our first successful attempt of such kind.

A primary concern on L2O models is that the neural network and training theories cannot guarantee convergence. Therefore, we follow the idea in [16] to wrap our method with a safeguard to ensure sufficient descent in objective value at each iteration.

2. Proposed method

Using a technique from [23], we derive an new parameterization, $\tau \in [0, 1]$, from BFGS and SR1 for the inverse Hessian:

$$H_{k+1}^{\text{BFGS-SR1}}(\tau) = \tau H_{k+1}^{\text{BFGS}} + (1-\tau) H_{k+1}^{\text{SR1}}. \tag{3}$$

By multiplying both sides of (3) by y_k, we can easily see that any $\tau \in (-\infty, +\infty)$ satisfies quasi-Newton condition since BFGS and SR1 satisfy the condition. When $\tau = \frac{y_k^T s_k}{y_k^T H_k y_k}$, (3) becomes a DFP update. Let us show that (3) is monotonic in parameter τ.

Lemma 1 *If $H_k \succcurlyeq A_k^{-1} \succ 0$, then, for any $\tau_1, \tau_2 \in \mathbb{R}$ such that $\tau_1 \geq \tau_2$, we have*

$$H_{k+1}^{\text{BFGS-SR1}}(\tau_1) \succcurlyeq H_{k+1}^{\text{BFGS-SR1}}(\tau_2).$$

Proof Without the loss of generality, suppose $H_k y_k \neq s_k$, then

$$\begin{aligned} H_{k+1}^{\text{BFGS-SR1}} = H_k &- \frac{(H_k y_k - s_k)(H_k y_k - s_k)^T}{(H_k y_k - s_k)^T y_k} \\ &+ \tau \left[\frac{(H_k y_k - s_k)(H_k y_k - s_k)^T}{(H_k y_k - s_k)^T y_k} - \frac{s_k y_k^T H_k + H_k y_k s_k^T}{y_k^T s_k} + \left(\frac{y_k^T H_k y_k}{y_k^T s_k} + 1 \right) \frac{s_k s_k^T}{y_k^T s_k} \right]. \end{aligned}$$

Denote $v_k \stackrel{\text{def}}{=} \frac{H_k y_k - s_k}{(H_k y_k - s_k)^T y_k} - \frac{s_k}{y_k^T s_k}$, then

$$H_{k+1}^{\text{BFGS-SR1}}(\tau_k) = H_k - \frac{(H_k y_k - s_k)(H_k y_k - s_k)^T}{(H_k y_k - s_k)^T y_k} + \tau (H_k y_k - s_k)^T y_k v_k v_k^T. \tag{4}$$

As $s_k = A_k^{-1} y_k$, if $H_k \succcurlyeq A_k^{-1}$, then $(H_k y_k - s_k)^T y_k > 0$. The claim now follows from the fact that $(H_k y_k - s_k)^T y_k v_k v_k^T \succcurlyeq 0$ when $H_k \succcurlyeq A_k^{-1}$. ■

We introduce our learning-based Broyden method with safegaurd (LBBS) in Algorithm 1. In lines 3 and 8, α_k' and τ_k' are the step size and Broyden parameter decided by two LSTMs, which share the same structure with independent trainable parameter sets, θ_α and θ_τ, respectively. The input parameter $z_k = \begin{bmatrix} x_k \\ \nabla f(x_k) \end{bmatrix}$ consists of the current iterate and gradient. The algorithm then determines the corresponding inverse Hessian approximation H_k', direction p_k', iterate x_{k+1}', gradient $\nabla f(x_{k+1}')$, and objective value $f(x_{k+1}')$. To improve stability, we propose a safeguard method inspired by [16]. In line 11, a safeguard checks if moving from x_k to x_{k+1}' satisfies the Wolfe conditions. If so, the current iteration is updated based on the learning model. Otherwise, in lines 14-17, the current iteration falls back to the BFGS approximator with a step size determined by the Wolfe line search. The safeguard in lines 11-17 can be removed to define an learning-based Broyden method (LBB) algorithm without safeguard.

The LSTM in line 3 is defined by

$$\text{LSTM}(z_k; \theta_\tau) = \sigma(c_k) o_k,$$

where

$$\begin{aligned} c_k &= g_k c_{k-1} + i_k \tilde{c}_k, \\ o_k &= \sigma(U_o z_{k+1} + w_o \, \tau_{k-1}), \\ g_k &= \sigma(U_g z_{k+1} + w_g \, \tau_{k-1}), \\ i_k &= \sigma(U_i z_{k+1} + w_i \, \tau_{k-1}), \\ \tilde{c}_k &= \tanh(U_{\tilde{c}} z_{k+1} + w_{\tilde{c}} \, \tau_{k-1}). \end{aligned}$$

Algorithm 1: Learning-based Broyden Method with Safeguard

Given x_0 and f, initialize $k \leftarrow 0$, determine $\nabla f(x_0)$, θ_τ and θ_α are obtained by ADAM[18] applied to solve the training model (5) approximately.

1. **while** x_k not converged **do**
2. **if** $k > 0$ **then**
3. $\tau'_k \leftarrow \mathrm{LSTM}(z_k; \theta_\tau)$
4. $H'_k \leftarrow \tau'_k H_k^{\mathrm{BFGS}} + (1 - \tau'_k) H_k^{\mathrm{SR1}}$
5. **else**
6. $H'_k \leftarrow I$
7. $p'_k \leftarrow -H'_k \nabla f(x_k)$
8. $\alpha'_k \leftarrow \mathrm{LSTM}(z_k; \theta_\alpha)$
9. propose $x'_{k+1} \leftarrow x_k + \alpha'_k p'_k$
10. determine $\nabla f(x'_{k+1})$ and $f(x'_{k+1})$
11. **if** $f(x'_{k+1}) \leq f(x_k) + c_1 \alpha'_k \nabla f^T(x_k) p'_k$ **and** $\nabla f^T(x'_{k+1}) p'_k \geq c_2 \nabla f^T(x_k) p'_k$ **then**
12. $x_{k+1} \leftarrow x'_{k+1}$, $\nabla f(x_{k+1}) \leftarrow \nabla f(x'_{k+1})$, and $H_k \leftarrow H'_k$
13. **else**
14. set $H_k \leftarrow H_k^{\mathrm{BFGS}}$ and $p_k \leftarrow -H_k \nabla f(x_k)$
15. determine α_k with Wolfe line search
16. $x_{k+1} \leftarrow x_k + \alpha_k p_k$
17. determine $\nabla f(x_{k+1})$
18. $k \leftarrow k + 1$
19. **end While**

Here, U_g, U_i, $U_{\tilde{c}}$, $U_o \in \mathbb{R}^{2n}$ and w_g, w_i, $w_{\tilde{c}}$, $w_o \in \mathbb{R}$ are trainable weights in θ_τ. The sigmoid and tanh activation functions are $\sigma(a) = \frac{1}{1+e^{-a}}$ and $\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$, respectively. An LSTM with the same structure but its own weights θ_α is applied to determine α'_k. During training, the safeguard is deactivated and the weights in the LSTMs are trained via

$$\underset{\theta_\tau, \theta_\alpha}{\text{minimize}}\, L(\theta_\tau, \theta_\alpha) = \sum_{k=1}^{K_{\text{train}}} \frac{f(x_k)}{f(x_0)}, \tag{5}$$

where $f(x_0)$ is the initial objective value and K_{train} is the number of training steps. We want the objective function value at the $1, 2, \cdots, K_{\text{train}}$ steps as small as possible.

3. Numerical Results

The proposed learning-based Broyden (LBB) model is trained on logistic regression with an l_2 regularizer, defined as:

$$\underset{x}{\text{minimize}}\, f(x) = -\frac{1}{m} \sum_{i=1}^{m} \left(b_i \log\, \sigma(A_i x) + (1 - b_i) \log\, (1 - \sigma(A_i x))\right) + \frac{\lambda}{n} \|x\|_2^2.$$

Here $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ are sample features and labels from EMNIST dataset [9], which contains 240,000 training images and 40,000 testing images with handwriting numbers from 0 to

9. We randomly extract 40,000 validation images from the training set with balanced classes. Each sample consists of 2 classes with $\frac{m}{2}$ images each, where each image is downsized to n features. For our experiments, we choose $m = 500$, $n = 100$, and $\lambda = 0.1$. LBB was trained to minimize (5) with $K_{\text{train}} = 10$ using ADAM[18] optimizer with a learning rate chosen by random search. we trained the model for 50 epochs where each epoch consists of 100 batches of training samples with a batch size of 64. LBB was compared with LBBS, DFP, BFGS, and SR1 on 100 testing samples with a maximum of 200 iterations and 1e-8 stopping tolerance. The Wolfe line search was implemented to determine appropriate step size for exact methods. We initialized $H_0 = I$ for all methods.

Figure 1 shows the result on a testing sample, where the optimal objective value $f(x^*)$ was determined by Newton's method with Wolfe line search. LBB requires a smaller number of n^2 computations to converge. The fluctuation of LBB is smoothed by the safeguard, so LBBS (S for safeguard) is the fastest to converge. SR1 fails to converge as it cannot maintain a positive definite H_k. Table 1 compares the average number of iterations and n^2 computations to converge across 100 testing samples. The statistics for SR1 and LBB are not shown there as they fail to converge within the maximum iterations on 11 and 3 samples, respectively. LBBS outperforms DFP and BFGS in both measures.

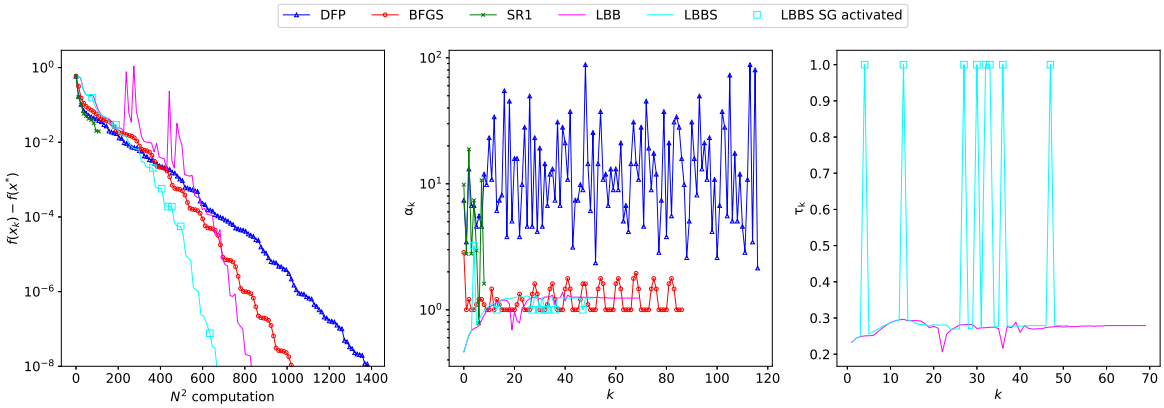

Figure 1: Objective error $f(x_k) - f(x_k^*)$ in the log scale (left), step size α_k for all methods (middle), and τ_k selected by LBB and LBBS (right). "$\tau_k = 1$" indicates when LBBS selects BFGS and activates the safeguard (lines 15-18).

Table 1: Average number of iterations and n^2 computations to converge across 100 testing samples.

Method	**Iteration**	n^2 **Computations**
DFP	106.0 ± 8.9	1271.5 ± 107.7
BFGS	80.6 ± 7.8	967.6 ± 93.1
LBBS	66.1 ± 31.7	845.3 ± 276.5

4. Conclusion

In this study, we present an L2O framework to select step size and Hessian approximator from a Broyden family. A safeguard stabilizes the selection and improves the performance. Our method exhibits consistently faster convergence in our preliminary experiment. Future work includes a rigorous convergence theory and various extensions toward larger problems and those other than logistic regression.

References

[1] Mehiddin Al-Baali. Variational quasi-newton methods for unconstrained optimization. *Journal of optimization theory and applications*, 77(1):127–143, 1993.

[2] Mehiddin Al-Baali. An efficient class of switching type algorithms in the broyden family. *Optimization Methods and Software*, 4(1):29–46, 1994.

[3] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016.

[4] Charles George Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970.

[5] Charles George Broyden, JE Dennis Jr, and Jorge J Moré. On the local and superlinear convergence of quasi-newton methods. *IMA Journal of Applied Mathematics*, 12(3):223–245, 1973.

[6] Richard H Byrd, Jorge Nocedal, and Ya-Xiang Yuan. Global convergence of a cass of quasi-newton methods on convex problems. *SIAM Journal on Numerical Analysis*, 24(5):1171–1190, 1987.

[7] Yue Cao, Tianlong Chen, Zhangyang Wang, and Yang Shen. Learning to optimize in swarms. In *Advances in Neural Information Processing Systems*, pages 15018–15028, 2019.

[8] Yutian Chen, Matthew W Hoffman, Sergio Gómez Colmenarejo, Misha Denil, Timothy P Lillicrap, Matt Botvinick, and Nando De Freitas. Learning to learn without gradient descent by gradient descent. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 748–756. JMLR. org, 2017.

[9] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters (2017). *arXiv preprint arXiv:1702.05373*.

[10] Jane Cullum and RK Brayton. Some remarks on the symmetric rank-one update. *Journal of Optimization Theory and Applications*, 29(4):493–519, 1979.

[11] William C Davidon. Variable metric method for minimization. *SIAM Journal on Optimization*, 1(1):1–17, 1991.

[12] Laurence Charles Ward Dixon. Variable metric algorithms: Necessary and sufficient conditions for identical behavior of nonquadratic functions. *Journal of Optimization Theory and Applications*, 10(1):34–40, 1972.

[13] Roger Fletcher. A new approach to variable metric algorithms. *The computer journal*, 13(3): 317–322, 1970.

[14] Roger Fletcher and Michael JD Powell. A rapidly convergent descent method for minimization. *The computer journal*, 6(2):163–168, 1963.

[15] Donald Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109):23–26, 1970.

[16] Howard Heaton, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Safeguarded learned convex optimization. *arXiv preprint arXiv:2003.01880*, 2020.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Jialin Liu, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Alista: Analytic weights are as good as learned weights in lista. *ICLR*, 2018.

[20] Michael David Powell. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In R. W. Cottle and C. E. Lemke, editors, *Nonlinear Programming, SIAM-AMS Proceedings*, Providence, RI, 1976. American Mathematical Society.

[21] MJD Powell. On the convergence of the variable metric algorithm. *IMA Journal of Applied Mathematics*, 7(1):21–36, 1971.

[22] MJD Powell. Some properties of the variable metric algorithm. *Numerical methods for nonlinear optimization*, pages 1–17, 1972.

[23] Anton Rodomanov and Yurii Nesterov. Greedy quasi-newton methods with explicit superlinear convergence. *arXiv preprint arXiv:2002.00657*, 2020.

[24] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[25] David F Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970.