**Les Cahiers du GERAD** ISSN: 0711–2440

# Stochastic damped L-BFGS with controlled norm of the Hessian approximation

S. Lotfi, T. Bonniot de Ruisselet, D. Orban, A. Lodi

G–2020–52

October 2020



La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2020
– Bibliothèque et Archives Canada, 2020

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2020
– Library and Archives Canada, 2020

**GERAD** HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

**Tél. : 514 340-6053**
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

# Stochastic damped L-BFGS with controlled norm of the Hessian approximation

**Sanae Lotfi** [a,b]
**Tiphaine Bonniot de Ruisselet** [a]
**Dominique Orban** [a]
**Andrea Lodi** [a,b]

[a] *GERAD, Montréal (Québec), Canada, H3T 2A7*

[a] *Department of Mathematics and Industrial Engineering, Polytechnique Montréal (Québec) Canada, H3C 3A7*

[b] *Canada Excellence Research Chair in Data Science for Real-Time Decision-Making, Polytechnique Montréal, Montréal (Québec), Canada, H3C 3A7*

sanae.lotfi@polymtl.ca
tiphaine.bonniot@gmail.com
dominique.orban@polymtl.ca
andrea.lodi@polymtl.ca

**October 2020**
**Les Cahiers du GERAD**
**G–2020–52**

**Abstract:** We propose a new stochastic variance-reduced damped L-BFGS algorithm, where we leverage estimates of bounds on the largest and smallest eigenvalues of the Hessian approximation to balance its quality and conditioning. Our algorithm, VARCHEN, draws from previous work that proposed a novel stochastic damped L-BFGS algorithm called SdLBFGS. We establish almost sure convergence to a stationary point and a complexity bound. We empirically demonstrate that VARCHEN is more robust than SdLBFGS-VR and SVRG on a modified DavidNet problem—a highly nonconvex and ill-conditioned problem that arises in the context of deep learning, and their performance is comparable on a logistic regression problem and a nonconvex support-vector machine problem.

# 1 Introduction and related work

We consider unconstrained stochastic minimization problems of $f : \mathbb{R}^n \to \mathbb{R}$ where

$$f(x) = \mathbb{E}_\xi[F(x,\xi)] \text{ (online)} \quad \text{or} \quad f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x) \text{ (finite sum)}, \tag{1}$$

where $\xi \in \mathbb{R}^d$ denotes a random variable, $F : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable and possibly nonconvex, $f_i$ is the loss corresponding to the $i$-th element of our dataset and $N$ is the size of the dataset. The algorithm developed below applies to both online and finite-sum problems.

Stochastic Gradient Descent (SGD) [5, 30] and its variants [12, 18, 25, 28, 32], including variance-reduced algorithms [13, 17, 26, 34], are widely used to solve (1) in machine learning. However, they might not be well-suited for highly nonconvex and ill-conditioned problems [6], which are more effectively treated using (approximate) second-order information. Second-order algorithms are well studied in the deterministic case [1, 9, 10, 11] but there are many areas to explore in the stochastic context that go beyond existing works [4, 7, 15, 24, 31]. Among these areas, the use of damping in L-BFGS is an interesting research direction to be leveraged in the stochastic case. Wang et al. [33] proposed a stochastic damped L-BFGS (SdLBFGS) algorithm and proved almost sure convergence to a stationary point. However, damping does not prevent the inverse Hessian approximation $H_k$ from being ill-conditioned [8]. The convergence of SdLBFGS may be heavily affected if the Hessian approximation becomes nearly singular during the iterations. In order to remedy this issue, Chen et al. [8] proposed to combine SdLBFGS with regularized BFGS [23]. Our approach differs.

## Our contributions:

- Less restrictive assumptions: we force $H_k$ to be uniformly bounded and positive definite by requiring the stochastic gradient to be Lipschitz continuous, which is a less restrictive assumption than those of Wang et al. [33], who require the stochastic function to be twice differentiable with respect to the parameter vector, and its Hessian to be bounded for all parameter and random sampling vectors.
- A new damped L-BFGS: we propose a new version of stochastic damped L-BFGS that maintains estimates of the smallest and largest eigenvalues of $H_k$. A solution is proposed when ill-conditioning is detected that preserves almost sure convergence to a stationary point.
- Choice of the initial inverse Hessian approximation: we propose a new formula for the initial inverse Hessian approximation to make the algorithm more robust to ill-conditioning.

**Notation** For a symmetric matrix $A$, we use $A \succ 0$ to indicate that $A$ is positive definite, and $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ to denote its smallest and largest eigenvalue, respectively. If $B$ is also symmetric, $B \preceq A$ means that $A - B$ is positive semidefinite. The identity matrix of appropriate size is denoted $I$. Finally, $\mathbb{E}_\xi[.]$ is the expectation over random variable $\xi$.

# 2 Formulation of our method

We assume that at iteration $k$, we can obtain a stochastic approximation

$$g(x_k, \xi_k) = \frac{1}{m_k}\sum_{i=1}^{m_k} \nabla f_{\xi_{k,i}}(x_k) \tag{2}$$

of $\nabla f(x_k)$, where $\xi_k$ denotes the subset of samples taken from a given set of realizations of $\xi$. The Hessian approximation constructed at iteration $k$ and its inverse are denoted by $B_k$ and $H_k$, respectively, such that $H_k = B_k^{-1}$ and $B_k \succ 0$. Iterates are updated according to

$$x_{k+1} = x_k + \alpha_k d_k, \quad \text{where} \quad d_k = -H_k g(x_k, \xi_k) \quad \text{and} \quad \alpha_k > 0 \text{ is the step size.} \tag{3}$$

The stochastic BFGS method [31] computes an updated approximation $H_{k+1}$ according to

$$H_{k+1} = V_k H_k V_k^\top + \rho_k s_k s_k^\top, \quad \text{where} \quad V_k = I - \rho_k s_k y_k^\top \quad \text{and} \quad \rho_k = 1/s_k^\top y_k, \tag{4}$$

which ensures that the secant equation $B_{k+1}s_k = y_k$ is satisfied, where

$$s_k = x_{k+1} - x_k, \quad \text{and} \quad y_k = g(x_{k+1}, \xi_k) - g(x_k, \xi_k). \tag{5}$$

If $H_k \succ 0$ and the curvature condition $s_k^\top y_k > 0$ holds, then $H_{k+1} \succ 0$ [see, for instance, 14].

Because storing $H_k$ and performing matrix-vector products is costly for large-scale problems, we use the limited-memory version of BFGS (L-BFGS) [22, 27], in which $H_k$ only depends on the most recent $p$ iterations and an initial $H_k^0 \succ 0$. The parameter $p$ is the *memory* of L-BFGS. The inverse Hessian update can be written as

$$\begin{aligned} H_k = (V_{k-1}^\top \dots V_{k-p}^\top) H_k^0 (V_{k-p} \dots V_{k-1}) + \\ \rho_{k-p}(V_{k-1}^\top \dots V_{k-p+1}^\top) s_{k-p} s_{k-p}^\top (V_{k-p+1} \dots V_{k-1}) + \dots + \rho_{k-1} s_{k-1} s_{k-1}^\top. \end{aligned} \tag{6}$$

When $\alpha_k$ in (3) is not computed using a Wolfe line search [35, 36], there is no guarantee that the curvature condition holds. A common strategy is to simply skip the update. By contrast, Powell [29] proposed *damping*, which consists in updating $H_k$ using a modified $y_k$, denoted by $\hat{y}_k$, to benefit from information discovered at iteration $k$ while ensuring sufficient positive definiteness. We use

$$\hat{y}_k := \theta_k y_k + (1 - \theta_k) B_{k+1}^0 s_k, \tag{7}$$

which is inspired by [33], and differs from the original proposal of Powell [29], where

$$\theta_k = 1 \ \text{ if } \ s_k^\top y_k \geq \eta s_k^\top B_{k+1}^0 s_k, \ \text{ and } \ (1-\eta)\frac{s_k^\top B_{k+1}^0 s_k}{s_k^\top B_{k+1}^0 s_k - s_k^\top y_k} \ \text{ otherwise,} \tag{8}$$

with $\eta \in (0,1)$ and $B_{k+1}^0 := (H_{k+1}^0)^{-1}$. The choice (7) ensures that the curvature condition

$$s_k^\top \hat{y}_k \geq \eta s_k^\top B_{k+1}^0 s_k \geq \eta \lambda_{\min}(B_{k+1}^0) \|s_k\|^2 > 0, \tag{9}$$

is always satisfied since $H_{k+1}^0 \succ 0$. We obtain the damped L-BFGS update, which is (6) with each $V_i$ and $\rho_i$ replaced with $\hat{V}_i = I - \hat{\rho}_i s_i^\top \hat{y}_i$ and $\hat{\rho}_i = 1/s_i^\top \hat{y}_i$.

## 3 A new stochastic damped L-BFGS with controlled Hessian norm

Our working assumption is

**Assumption 1** *There is $\kappa_{low} \in \mathbb{R}$ such that $f(x) \geq \kappa_{low}$ for all $x \in \mathbb{R}^n$, $f$ is $\mathcal{C}^1$ over $\mathbb{R}^n$, and there is $L > 0$ such that for all $x$, $y \in \mathbb{R}^n$, $\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|$.*

We begin by deriving bounds on the smallest and largest eigenvalues of $H_{k+1}$ as functions of bounds on those of $H_k$. Proofs can be found in Section A.

**Lemma 1** *Let $s$ and $y \in \mathbb{R}^n$ such that $s^\top y \geq \gamma \|s\|^2$ with $\gamma > 0$, and such that $\|y\| \leq L_y \|s\|$, with $L_y > 0$. Let $A = \mu V V^\top + \rho s s^\top$, where $\rho = 1/s^\top y$, $\mu > 0$, and $V = I - \rho s y^\top$. Then,*

$$0 < \min\left(\frac{1}{L_y}, \frac{\mu}{1 + \frac{\mu}{\gamma} L_y^2}\right) \leq \lambda_{\min}(A) \leq \lambda_{\max}(A) \leq \frac{1}{\gamma} + \max\left(0, \frac{\mu}{\gamma^2} L_y^2 - \frac{\mu}{1 + \frac{\mu}{\gamma} L_y^2}\right).$$

To use Lemma 1 to obtain bounds on the eigenvalues of $H_{k+1}$, we make the following assumption:

**Assumption 2** *There is $L_g > 0$ such that for all $x$, $y \in \mathbb{R}^n$, $\|g(x,\xi) - g(y,\xi)\| \leq L_g \|x - y\|$.*

Assumption 2 is required to prove convergence and convergence rates for most recent stochastic quasi-Newton methods [37]. It is less restrictive than requiring $f(x,\xi)$ to be twice differentiable with respect to $x$, and the Hessian $\nabla^2_{xx} f(x,\xi)$ to be bounded for any $x$, $\xi$, as in [33].

The next theorem shows that the eigenvalues of $H_{k+1}$ are bounded and bounded away from zero.

**Theorem 1** *Let Assumptions 1 and 2 hold. Let $H^0_{k+1} \succ 0$ and $p > 0$. If $H_{k+1}$ is obtained by applying $p$ times the damped BFGS update formula with inexact gradient to $H^0_{k+1}$, there exist easily computable constants $\lambda_{k+1}$ and $\Lambda_{k+1}$ that depend on $L_g$ and $H^0_{k+1}$ such that $0 < \lambda_{k+1} \leq \lambda_{\min}(H_{k+1}) \leq \lambda_{\max}(H_{k+1}) \leq \Lambda_{k+1}$.*

The precise form of $\lambda_{k+1}$ and $\Lambda_{k+1}$ is given in Section A.

A common choice for $H^0_{k+1}$ is $H^0_{k+1} = \gamma^{-1}_{k+1} I$ where $\gamma_{k+1} = y_k^\top y_k / s_k^\top y_k$, is the *scaling parameter*. This choice ensures that the search direction is well scaled, which promotes large steps. To keep $H^0_{k+1}$ from becoming nearly singular or non positive definite, we define

$$H^0_{k+1} = \Big(\max(\underline{\gamma}_{k+1}, \min(\gamma_{k+1}, \overline{\gamma}_{k+1}))\Big) I, \tag{10}$$

where $0 < \underline{\gamma}_{k+1} < \overline{\gamma}_{k+1}$ can be constants or iteration dependent.

The Hessian-gradient product used to compute the search direction $d_k = -H_k g(x_k, \xi_k)$ can be obtained cheaply by exploiting a recursive algorithm [27], as described in Algorithm 2 in Section B.

Motivated by the success of recent methods combining variance reduction with stochastic L-BFGS [15, 24, 33], we apply an SVRG-like type of variance reduction [17] to the update. Not only would this accelerate the convergence, since we can choose a constant step size, but it also improves the quality of the curvature approximation.

We summarize our complete algorithm, VAriance-Reduced stochastic damped L-BFGS with Controlled HEssian Norm (VARCHEN), as Algorithm 1.

**Algorithm 1** Variance-Reduced Stochastic Damped L-BFGS with Controlled Hessian Norm

1: Choose $x_0 \in \mathbb{R}^n$, step size sequence $\{\alpha_k > 0\}_{k\geq 0}$, batch size sequence $\{m_k > 0\}_{k\geq 0}$, eigenvalue limits $\lambda_{\max} > \lambda_{\min} > 0$, memory parameter $p$, total number of epochs $N_{\text{epochs}}$, and sequences $\{\underline{\gamma}_k > 0\}_{k\geq 0}$ and $\{\overline{\gamma}_{k+1} > 0\}_{k\geq 0}$, such that $0 < \lambda_{\min} < \underline{\gamma}_k < \overline{\gamma}_k < \lambda_{\max}$, for every $k \geq 0$. Set $k = 0$ and $H_0 = I$.
2: **for** $t = 1, \ldots, N_{\text{epochs}}$ **do**
3: Define $x^t_k = x_k$ and compute the full gradient $\nabla f(x^t_k)$. Set $M = 0$.
4: **while** $M < N$ **do**
5: Sample batch $\xi_k$ of size $m_k \leq N - M$ and compute $g(x_k, \xi_k)$ and $g(x^t_k, \xi_k)$.
6: Define $\tilde{g}(x_k, \xi_k) = g(x_k, \xi_k) - g(x^t_k, \xi_k) + \nabla f(x^t_k)$.
7: Estimate $\Lambda_k$ and $\lambda_k$ in Theorem 1. If $\Lambda_k > \lambda_{\max}$ or $\lambda_k < \lambda_{\min}$, delete $s_i$, $y_i$ and $\hat{y}_i$ for $i = k - p + 1, \ldots, k - 2$.
8: Compute $d_k = -H_k \tilde{g}(x_k, \xi_k)$.
9: Define $x_{k+1} = x_k + \alpha_k d_k$, and compute $s_k$, $y_k$ as in (5), and $\hat{y}_k$ as in (7).
10: Increment $k$ by one and update $M \leftarrow M + m_k$.

In step 7 of Algorithm 1, we compute an estimate of the upper and lower bounds on $\lambda_{\max}(H_k)$ and $\lambda_{\min}(H_k)$, respectively. The only unknown quantity in the expressions of $\Lambda_k$ and $\lambda_k$ in Theorem 1 is $L_g$, which we estimate as $L_g \approx L_{g,k} := \|y_k\|/\|s_k\|$. When the estimates are not within the limits $[\lambda_{\min}, \lambda_{\max}]$, we delete $s_i$, $y_i$ and $\hat{y}_i$, $i \in \{k - p + 1, \ldots, k - 2\}$ from storage, such that $H_k g(x_k, \xi_k)$ is computed using the most recent pair $(s_{k-1}, \hat{y}_{k-1})$ only and $d_k = -H_k g(x_k, \xi_k)$. Finally, a full gradient is computed once in every epoch in step 3. The term $g(x^t_k, \xi_k) - \nabla f(x^t_k)$ can be seen as the bias in the gradient estimation $g(x_k, \xi_k)$, and it is used here to correct the gradient approximation in step 6.

## 4 Convergence and complexity analysis

We show that Algorithm 1 satisfies the assumptions of the convergence analysis and iteration complexity of Wang et al. [33] for stochastic quasi-Newton methods. We make an additional assumption used by Wang et al. [33] to establish global convergence.

**Assumption 3** *For all $k$, $\xi_k$ is independent of $\{x_1, \ldots, x_k\}$, $\mathbb{E}_{\xi_k}[g(x_k, \xi_k)] = \nabla f(x_k)$, and there exists $\sigma > 0$ such that $\mathbb{E}_{\xi_k}[\|g(x_k, \xi_k) - \nabla f(x_k)\|^2] \leq \sigma^2$.*

Our first result follows from Wang et al. [33, Theorem 2.6], whose remaining assumptions are satisfied as a consequence of (2), Theorem 1, the mechanism of Algorithm 1, (4) and our choice of $\alpha_k$ below.

**Theorem 2** *Assume $m_k = m$ for all $k$, that Assumptions 1 to 3 hold for $\{x_k\}$ generated by Algorithm 1, and that $\alpha_k := c/(k+1)$ where $0 < c \leq \lambda_{\min}/(L\lambda_{\max})$. Then, $\liminf \|\nabla f(x_k)\| = 0$ with probability 1. Moreover, there is $M_f > 0$ such that $\mathbb{E}[f(x_k)] \leq M_f$ for all $k$. If we additionally assume that there exists $M_g > 0$ such that $\mathbb{E}_{\xi_k}[\|g(x_k, \xi_k)\|^2] \leq M_g$, then $\lim \|\nabla f(x_k)\| = 0$ with probability 1.*

Our next result follows in the same way from Wang et al. [33, Theorem 2.8].

**Theorem 3** *Under the assumptions of Theorem 2, if $\alpha_k = \lambda_{\min}/(L\lambda_{\max}^2)k^{-\beta}$ for all $k > 0$, with $\beta \in (\frac{1}{2}, 1)$, then, for any $\epsilon \in (0, 1)$, after at most $T = \mathcal{O}(\epsilon^{-1/(1-\beta)})$ iterations, we achieve*

$$\frac{1}{T}\sum_{k=1}^{T} \mathbb{E}\left[\|\nabla f(x_k)\|^2\right] \leq \epsilon. \tag{11}$$

## 5 Experimental results

We compare VARCHEN to SdLBFGS-VR [33] and to SVRG [17] for solving a multi-class classification problem. We train a modified version[1] of the deep neural network model DavidNet[2] proposed by David C. Page, on CIFAR-10 [19] for 20 epochs. Note that we also used VARCHEN and SdLBFGS-VR to solve a logistic regression problem using the MNIST dataset [20] and a nonconvex support-vector machine problem with a sigmoid loss function using the RCV1 dataset [21]. The performance of both algorithms are on par on those problems because, in contrast with DavidNet on CIFAR-10, they are not highly nonconvex or ill conditioned.

Figure 1 shows that VARCHEN outperforms SdLBFGS-VR for the training loss minimization task, and both outperform SVRG. VARCHEN has an edge over SdLBFGS-VR in terms of the validation accuracy, and both outperform SVRG. More importantly, the performance of VARCHEN is more consistent than that of SdLBFGS-VR, displaying a smoother, less oscillatory behaviour. To further investigate this observation, we plot the evolution of $\Lambda_k$ and $\lambda_k$ as shown in Figure 2. We see that the estimate of the lower bound on the smallest eigenvalue is smaller for SdLBFGS-VR compared to VARCHEN. We also notice that the estimate of the upper bound of the largest eigenvalue of $H_k$ takes even more extreme values for SdLBFGS-VR compared to VARCHEN. The extreme values $\lambda_k$ and $\Lambda_k$ reflect an ill-conditioning problem encountered when using SdLBFGS-VR and we believe that it explains the extreme oscillations in the performance of SdLBFGS-VR.

[1] FastResNet Hyperparameters tuning with Ax on CIFAR10

[2] https://myrtle.ai/learn/how-to-train-your-resnet-4-architecture/

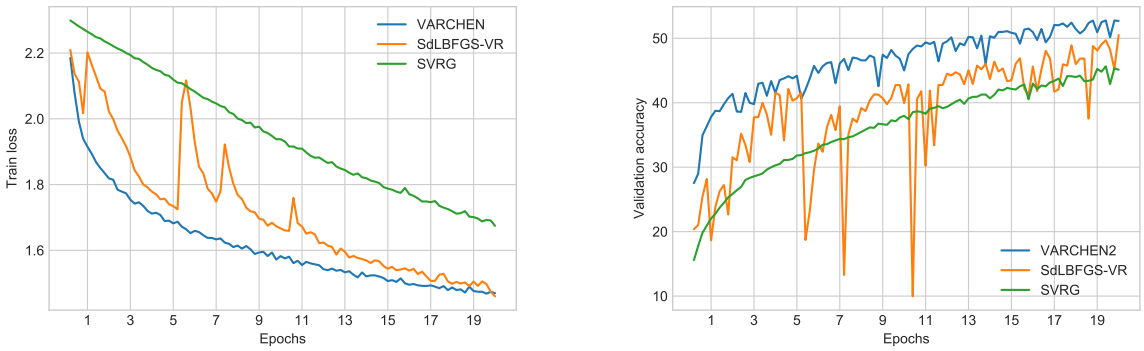


**Figure 1: Evolution of the training loss (left) and the validation accuracy (right) for training a modified DavidNet on CIFAR-10.**

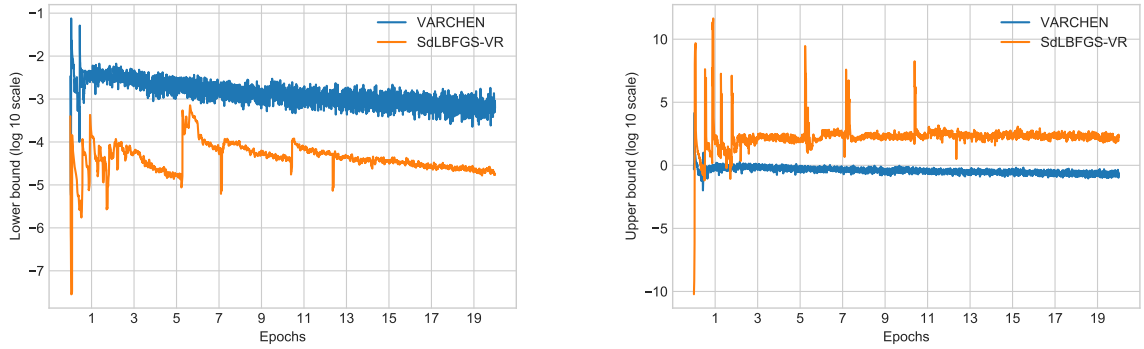


**Figure 2: Evolution of the lower bound on the smallest eigenvalue $\lambda_k$ (left) and the upper bound on the largest eigenvalue $\Lambda_k$ (right) on a base 10 logarithmic scale for training a modified DavidNet on CIFAR-10.**

# 6 Conclusion

We used the stochastic damped L-BFGS algorithm in a nonconvex setting, where there are no guarantees that $H_k$ remains well-conditioned and numerically nonsingular throughout. We introduced a new stochastic damped L-BFGS algorithm that monitors the quality of $H_k$ during the optimization by maintaining bounds on its largest and smallest eigenvalues. Our work is the first to address the Hessian singularity problem by approximating and leveraging such bounds. Moreover, we proposed a new initial inverse Hessian approximation that results in a smoother, less oscillatory training loss and validation accuracy evolution. Additionally, we used variance reduction in order to improve the quality of the curvature approximation and accelerate convergence. Our algorithm converges almost-surely to a stationary point and numerical experiments have shown that it is more robust to ill-conditioned problems and more suitable to the highly nonconvex context of deep learning than SdLBFGS-VR. We consider this work to be a first step towards the use of bounds estimates to control the quality of the Hessian approximation in approximate second-order algorithms. Future work should aim to improve the quality of these bounds and explore another form of variance reduction that consists of adaptive sampling [2, 3, 16].

# Appendix A Proofs

**Proof of Lemma 1.** First notice that

$$\gamma\|s\|^2 \leq s^\top y \leq \|s\|\|y\|, \quad \text{and thus} \quad \|s\| \leq \frac{1}{\gamma}\|y\|.$$

Therefore,

$$\frac{1}{\|s\|\|y\|} \leq \rho \leq \frac{1}{\gamma}\frac{1}{\|s\|^2}. \tag{12}$$

Since $A$ is a real symmetric matrix, the spectral theorem states that its eigenvalues are real and it can be diagonalized by an orthogonal matrix. That means that we can find $n$ orthogonal eigenvectors and $n$ eigenvalues counted with multiplicity.

Consider first the special case where $s$ and $y$ are collinear, i.e. there exists $\theta > 0$ such that $y = \theta s$. Any vector such that $u \in s^\perp$, where $s^\perp = \{x \in \mathbb{R}^n : x^\top s = 0\}$, is an eigenvector of $A$ associated with the eigenvalue $\mu$ of multiplicity $n-1$. Moreover, $s^\top y = \theta\|s\|^2 = \|s\|\|y\|$, $\rho = 1/(\theta\|s\|^2)$ and we have

$$As = \left[\mu\left(I - \rho\theta ss^\top\right)^2 + \rho ss^\top\right] s = \left[\mu\left(1 - \rho\theta\|s\|^2\right)^2 + \rho\|s\|^2\right] s = \rho\|s\|^2 s.$$

Let us call $\lambda = \rho\|s\|^2$, the eigenvalue associated with eigenvector $s$. From (12) and $\|y\| \leq L_y\|s\|$, we deduce that

$$\frac{1}{L_y} \leq \lambda \leq \frac{1}{\gamma}.$$

Suppose now that $s$ and $y$ are linearly independent. Any $u$ such that $u^\top s = 0 = u^\top y$ satisfies $Au = \mu u$. This provides us with a $(n-2)$-dimensional eigenspace $S$, associated to the eigenvalue $\mu$ of multiplicity $n-2$. Note that

$$\begin{aligned} As &= \rho\|s\|^2\ (1 + \mu\rho\|y\|^2)s - \rho\|s\|^2\mu y, \\ Ay &= s. \end{aligned}$$

Thus neither $s$ nor $y$ is an eigenvector of $A$. Now consider an eigenvalue $\lambda$ associated with an eigenvector $u$, such that $u \in S^\perp$. Since $s$ and $y$ are linearly-independent, we can search for $u$ of the form $u = s + \beta y$ with $\beta > 0$. The condition $Au = \lambda u$ yields

$$\begin{aligned} \rho\|s\|^2\ (1 + \mu\rho\|y\|^2) + \beta &= \lambda, \\ -\rho\|s\|^2\mu &= \lambda\beta. \end{aligned}$$

We eliminate $\beta = \lambda - \rho\|s\|^2\ (1 + \mu\rho\|y\|^2)$ and obtain

$$p(\lambda) = 0,$$

where

$$p(\lambda) = \lambda^2 - \lambda\rho\|s\|^2\ (1 + \mu\rho\|y\|^2) + \rho\|s\|^2\mu.$$

The roots of $p$ must be the two remaining eigenvalues $\lambda_1 \leq \lambda_2$ that we are looking for. In order to establish the lower bound, we need a lower bound on $\lambda_1$ whereas to establish the upper bound, we need an upper bound on $\lambda_2$.

On the one hand, let $l$ be the tangent to the graph of $p$ at $\lambda = 0$, defined by

$$l(\lambda) = p(0) + p'(0)\lambda = \mu\rho\|s\|^2 - \lambda\rho\|s\|^2\left(1 + \mu\rho\|y\|^2\right).$$

Its unique root is

$$\bar{\lambda} = \frac{\mu}{1 + \mu\rho\|y\|^2}.$$

From (12) and since $\|y\| \leq L_y\|s\|$, we deduce that

$$\bar{\lambda} \geq \frac{\mu}{1 + \frac{\mu}{\gamma}\frac{\|y\|^2}{\|s\|^2}} \geq \frac{\mu}{1 + \frac{\mu}{\gamma}L_y^2}.$$

Since $p$ is convex, it remains above its tangent, and $\bar{\lambda} \leq \lambda_1$.

Finally,

$$\lambda_{\min}(A) \geq \min\left(\frac{1}{L_y}, \frac{\mu}{1 + \frac{\mu}{\gamma}L_y^2}\right) > 0.$$

This establishes the lower bound.

On the other hand, the discriminant $\Delta = \rho^2\|s\|^4(1 + \mu\rho\|y\|^2)^2 - 4\rho\|s\|^2\mu$ must be nonnegative since $A$ is real symmetric, and its eigenvalues are real. We have

$$\lambda_2 = \frac{\rho\|s\|^2(1 + \mu\rho\|y\|^2) + \sqrt{\rho^2\|s\|^4(1 + \mu\rho\|y\|^2)^2 - 4\rho\|s\|^2\mu}}{2}.$$

For any positive $a$ and $b$ such that $a^2 - b > 0$, we have $\sqrt{a^2 - b} \leq a - \frac{b}{2a}$. Thus,

$$\lambda_2 \leq \rho\|s\|^2(1 + \mu\rho\|y\|^2) - \frac{\mu}{1 + \mu\rho\|y\|^2}.$$

From (12), we deduce that

$$\lambda_2 \leq \frac{1}{\gamma} + \frac{\mu}{\gamma^2}\frac{\|y\|^2}{\|s\|^2} - \frac{\mu}{1 + \frac{\mu}{\gamma}\frac{\|y\|^2}{\|s\|^2}}.$$

And since $\|y\| \leq L_y\|s\|$, it follows

$$\lambda_2 \leq \frac{1}{\gamma} + \frac{\mu}{\gamma^2}L_y^2 - \frac{\mu}{1 + \frac{\mu}{\gamma}L_y^2}.$$

Finally,

$$\lambda_{\max}(A) \leq \max\left(\frac{1}{\gamma}, \frac{1}{\gamma} + \frac{\mu}{\gamma^2}L_y^2 - \frac{\mu}{1 + \frac{\mu}{\gamma}L_y^2}\right),$$

which establishes the upper bound. □

**Proof of Theorem 1.** Consider one damped BFGS update using $s$ and $y$ defined in (5) and $\hat{y}_k$ defined in (7), i.e, $p = 1$,

$$H_{k+1} = \hat{V}_k H_{k+1}^0 \hat{V}_k^\top + \hat{\rho}_k s_k s_k^\top, \quad \text{where} \quad \hat{\rho}_k = 1/s_k^\top \hat{y}_k, \quad \hat{V}_k = I - \hat{\rho}_k s_k \hat{y}_k^\top.$$

Let $0 < \mu_1 := \lambda_{\min}(H_{k+1}^0) \leq \mu_2 := \lambda_{\max}(H_{k+1}^0)$. We have

$$\lambda_{\min}(\mu_1 \hat{V}_k \hat{V}_k^\top + \hat{\rho}_k s_k s_k^\top) \leq \lambda_{\min}(H_{k+1}) \leq \lambda_{\max}(H_{k+1}) \leq \lambda_{\max}(\mu_2 \hat{V}_k \hat{V}_k^\top + \hat{\rho}_k s_k s_k^\top). \tag{13}$$

Let us show that we can apply Lemma 1 to

$$A_1 := \mu_1 \hat{V}_k \hat{V}_k^\top + \hat{\rho}_k s_k s_k^\top \quad \text{and} \quad A_2 := \mu_2 \hat{V}_k \hat{V}_k^\top + \hat{\rho}_k s_k s_k^\top.$$

From (9), we obtain

$$s_k^\top \hat{y}_k \geq \eta \lambda_{\min}(B_{k+1}^0)\|s_k\|^2 = \frac{\eta}{\lambda_{\max}(H_{k+1}^0)}\|s_k\|^2 = \frac{\eta}{\mu_2}\|s_k\|^2.$$

Assumption 2 yields

$$\|\hat{y}_k\| = \|\theta_k y_k + (1-\theta_k)B_{k+1}^0 s_k\| \leq \|y_k\| + \|B_{k+1}^0 s_k\| \leq (L_g + 1/\mu_1)\|s_k\|.$$

Therefore, we can first apply Lemma 1 with $s_k$, $\hat{y}_k$, $\gamma = \eta/\mu_2 > 0$, $L_y = L_g + 1/\mu_1 > 0$ and $\mu = \mu_1 > 0$ for $A_1$, and apply it again with $\mu = \mu_2 > 0$ for $A_2$. Let $L_1 := L_g + 1/\mu_1$. Lemma 1 and (13) yield

$$\lambda_{\min}(H_{k+1}) \geq \min\left(\frac{1}{L_1}, \frac{\mu_1}{1 + \frac{\mu_1\mu_2}{\eta}L_1^2}\right) > 0,$$
$$\lambda_{\max}(H_{k+1}) \leq \frac{\mu_2}{\eta} + \max\left(0, \frac{\mu_2^3}{\eta^2}L_1^2 - \frac{\mu_2}{1 + \frac{\mu_2^2}{\eta}L_1^2}\right).$$

Now, consider the case where $p > 1$ and let

$$H_{k+1}^{(h+1)} := \hat{V}_{k-h} H_{k+1}^{(h)} \hat{V}_{k-h}^\top + \hat{\rho}_{k-h} s_{k-h} s_{k-h}^\top, \quad 0 \leq h \leq p-1,$$

where

$$H_{k+1}^{(p)} := H_{k+1}, \quad \hat{\rho}_{k-h} = 1/s_{k-h}^\top \hat{y}_{k-h}, \quad \hat{V}_k = I - \hat{\rho}_{k-h} s_{k-h} \hat{y}_{k-h}^\top.$$

Similarly to the case $p = 1$, we may write

$$\lambda_{\min}(H_{k+1}^{(h+1)}) \geq \lambda_{\min}(\mu_1^{(h)} \hat{V}_{k-h}\hat{V}_{k-h}^\top + \hat{\rho}_{k-h} s_{k-h} s_{k-h}^\top), \qquad \mu_1^{(h)} := \lambda_{\min}(H_{k+1}^{(h)}),$$
$$\lambda_{\max}(H_{k+1}^{(h+1)}) \leq \lambda_{\max}(\mu_2^{(h)} \hat{V}_{k-h}\hat{V}_{k-h}^\top + \hat{\rho}_{k-h} s_{k-h} s_{k-h}^\top), \qquad \mu_2^{(h)} := \lambda_{\max}(H_{k+1}^{(h)}).$$

Assume by recurrence that $0 < \mu_1^{(h)} \leq \mu_2^{(h)}$. We show that we can apply Lemma 1 to

$$A_1^{(h)} := \mu_1^{(h)} \hat{V}_{k-h}\hat{V}_{k-h}^\top + \hat{\rho}_{k-h} s_{k-h} s_{k-h}^\top \quad \text{and} \quad A_2^{(h)} := \mu_2^{(h)} \hat{V}_{k-h}\hat{V}_{k-h}^\top + \hat{\rho}_{k-h} s_{k-h} s_{k-h}^\top.$$

From (9), we have

$$s_{k-h}^\top \hat{y}_{k-h} \geq \eta \lambda_{\min}(B_{k-h+1}^0)\|s_{k-h}\|^2 = \frac{\eta}{\lambda_{\max}(H_{k-h+1}^0)}\|s_{k-h}\|^2.$$

Using Assumption 2,

$$\|\hat{y}_{k-h}\| = \|\theta_{k-h} y_{k-h} + (1-\theta_{k-h})B_{k-h+1}^0 s_{k-h}\| \leq L_g\|s_{k-h}\| + \|B_{k-h+1}^0 s_{k-h}\|,$$

so that

$$\|\hat{y}_{k-h}\| \leq (L_g + \frac{1}{\lambda_{\min}(H_{k-h+1}^0)})\|s_{k-h}\|.$$

We first apply Lemma 1 with $s = s_{k-h}$, $y = \hat{y}_{k-h}$, $\gamma = \eta/\lambda_{\max}(H_{k-h+1}^0) > 0$, $L_y = L_g + 1/\lambda_{\min}(H_{k-h+1}^0) > 0$ and $\mu = \mu_1^{(h)} > 0$ for $A_1^{(h)}$, and apply it a second time with $\mu = \mu_2^{(h)} > 0$ for $A_2^{(h)}$. Let $L_{k-h+1} := L_g + 1/\lambda_{\min}(H_{k-h+1}^0)$ and $\gamma_{k-h+1} = \eta/\lambda_{\max}(H_{k-h+1}^0)$. Then we have

$$\lambda_{\min}(H_{k+1}^{(h+1)}) \geq \min\left(\frac{1}{L_{k-h+1}}, \frac{\lambda_{\min}(H_{k+1}^{(h)})}{1 + \frac{\lambda_{\min}(H_{k+1}^{(h)})}{\gamma_{k-h+1}}(L_{k-h+1})^2}\right), \tag{14}$$

and

$$\lambda_{\max}(H_{k+1}^{(h+1)}) \leq \frac{1}{\gamma_{k-h+1}} + \max\left(0, \frac{\lambda_{\max}(H_{k+1}^{(h)})}{\gamma_{k-h+1}^2} L_{k-h+1}^2 - \frac{\lambda_{\max}(H_{k+1}^{(h)})}{1 + \frac{\lambda_{\max}(H_{k+1}^{(h)})}{\gamma_{k-h+1}} L_{k-h+1}^2}\right). \tag{15}$$

It is clear that we can obtain the lower bound on $\lambda_{\min}(H_{k+1})$ recursively using (14). Obtaining the upper bound on $\lambda_{\max}(H_{k+1})$ using (15) is trickier. However, we notice that inequality (15) implies

$$\lambda_{\max}(H_{k+1}^{(h+1)}) \leq \frac{1}{\gamma_{k-h+1}} + \max\left(0, \frac{\lambda_{\max}(H_{k+1}^{(h)})}{\gamma_{k-h+1}^2} L_{k-h+1}^2 - \frac{\lambda_{\min}(H_{k+1}^{(h)})}{1 + \frac{\lambda_{\max}(H_{k+1}^{(h)})}{\gamma_{k-h+1}} L_{k-h+1}^2}\right).$$

This upper bound is less tight but it allows us to bound $\lambda_{\max}(H_{k+1})$ recursively. □

# Appendix B Algorithms

The Two-loop recursion algorithm for evaluating the Hessian-gradient product is given by Algorithm 2.

**Algorithm 2** Two-loop recursion algorithm for Hessian-gradient product computation

**Require:** Current iterate $x_k$, $g(x_k, \xi_k)$, $\hat{\rho}_i = 1/s_i^\top \hat{y}_i$, $s_i$, $\hat{y}_i$ for $i \in \{k-p, \ldots, k-1\}$.
1: Define $g = g(x_k, \xi_k)$
2: **for** $i = k-1, k-2, \ldots, k-p$ **do**
3: Compute $\nu_i = \hat{\rho}_i s_i^\top g$
4: Compute $g = g - \nu_i \hat{y}_i$
5: Compute $q = H_k^0 g$ using (10).
6: **for** $i = k-p, k-p+1, \ldots, k-1$ **do**
7: Compute $\mu = \hat{\rho}_i \hat{y}_i^\top q$
8: Compute $q = q + (\nu_i - \mu) s_i$
9: Return $q = H_k\, g(x_k, \xi_k)$

# Appendix C Experimental Setting

CIFAR-10 is a dataset that contains 60,000 colour images with labels in 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck), each containing 6,000 images. We use 50,000 images for the training task and 10,000 images for the validation task.

Details of the experiments:

- We apply our Hessian norm control using the bound on the maximum and the minimum eigenvalues of $H_{k+1}$, where the latter is equivalent to controlling $\|B_{k+1}\|$;
- In the definition of $H_{k+1}^0$ in (10), we choose $\underline{\gamma}_{k+1}$ and $\overline{\gamma}_{k+1}$ constant;
- The numerical values for all algorithms are the ones that yielded the best results among all sets of values that we experimented with.

Numerical values:

- For all algorithms: we train the network for 20 epochs and use a batch size of 256 samples;
- For SVRG, we choose a step size equal to 0.001;
- For both SdLBFGS-VR and Algorithm 1, the memory parameter $p = 10$, the minimal scaling parameter $\underline{\gamma}_{k+1} = 0.1$ for all $k$, the constant step size $\alpha_k = 0.1$ and $\eta = 0.25$;
- For Algorithm 1, we use a maximal scaling parameter $\overline{\gamma}_{k+1} = 10^5$ for all $k$, a lower bound limit $\lambda_{\min} = 10^{-5}$ and an upper bound limit $\lambda_{\max} = 10^5$.

## References


[1] Shun-Ichi Amari. Natural gradient works efficiently in learning. Neural computation, 10(2):251–276, 1998.

[2] Raghu Bollapragada and Stefan M Wild. Adaptive sampling quasi-Newton methods for derivative-free stochastic optimization. arXiv preprint arXiv:1910.13516, 2019.

[3] Raghu Bollapragada, Dheevatsa Mudigere, Jorge Nocedal, Hao-Jun Michael Shi, and Ping Tak Peter Tang. A progressive batching l-bfgs method for machine learning. arXiv preprint arXiv:1802.05374, 2018.

[4] Antoine Bordes, Léon Bottou, and Patrick Gallinari. SGD-QN: Careful quasi-Newton stochastic gradient descent. Journal of Machine Learning Research, 10(Jul):1737–1754, 2009.

[5] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010, pages 177–186. Springer, 2010.

[6] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. Siam Review, 60(2):223–311, 2018.

[7] Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-Newton method for large-scale optimization. SIAM Journal on Optimization, 26(2):1008–1031, 2016.

[8] Huiming Chen, Ho-Chun Wu, Shing-Chow Chan, and Wong-Hing Lam. A stochastic quasi-Newton method for large-scale nonconvex optimization with applications. IEEE Transactions on Neural Networks and Learning Systems, 2019.

[9] Ron S Dembo, Stanley C Eisenstat, and Trond Steihaug. Inexact Newton methods. SIAM Journal on Numerical analysis, 19(2):400–408, 1982.

[10] John E Dennis and Jorge J Moré. A characterization of superlinear convergence and its application to quasi-Newton methods. Mathematics of computation, 28(126):549–560, 1974.

[11] John E Dennis Jr and Robert B Schnabel. Numerical methods for unconstrained optimization and nonlinear equations, volume 16. SIAM, Philadelphia, PA, 1996.

[12] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12(7), 2011.

[13] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In Advances in Neural Information Processing Systems, pages 689–699, 2018.

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

[15] Robert Gower, Donald Goldfarb, and Peter Richtárik. Stochastic block BFGS: Squeezing more curvature out of data. In International Conference on Machine Learning, pages 1869–1878, 2016.

[16] Afrooz Jalilzadeh, Angelia Nedić, Uday V Shanbhag, and Farzad Yousefian. A variable sample-size stochastic quasi-Newton method for smooth and nonsmooth stochastic convex optimization. In 2018 IEEE Conference on Decision and Control (CDC), pages 4097–4102. IEEE, 2018.

[17] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In Advances in neural information processing systems, pages 315–323, 2013.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In International Conference on Learning Representations, 2015.

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[20] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist, 2, 2010.

[21] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. Journal of machine learning research, 5(Apr):361–397, 2004.

[22] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. Mathematical programming, 45(1-3):503–528, 1989.

[23] Aryan Mokhtari and Alejandro Ribeiro. Res: Regularized stochastic BFGS algorithm. IEEE Transactions on Signal Processing, 62(23):6089–6104, 2014.

[24] Philipp Moritz, Robert Nishihara, and Michael Jordan. A linearly-convergent stochastic L-BFGS algorithm. In Artificial Intelligence and Statistics, pages 249–258, 2016.

[25] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. In Dokl. akad. nauk Sssr, volume 269, pages 543–547, 1983.

[26] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. arXiv preprint arXiv:1703.00102, 2017.

[27] Jorge Nocedal. Updating quasi-Newton matrices with limited storage. Mathematics of computation, 35 (151):773–782, 1980.

[28] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics, 4(5):1–17, 1964.

[29] Michael JD Powell. Algorithms for nonlinear constraints that use Lagrangian functions. Mathematical programming, 14(1):224–248, 1978.

[30] Herbert Robbins and Sutton Monro. A stochastic approximation method. The annals of mathematical statistics, pages 400–407, 1951.

[31] Nicol N Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-Newton method for online convex optimization. In Artificial intelligence and statistics, pages 436–443, 2007.

[32] Tijmen Tieleman and Geoffrey Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. COURSERA Neural Networks Mach. Learn, 2012.

[33] Xiao Wang, Shiqian Ma, Donald Goldfarb, and Wei Liu. Stochastic quasi-Newton methods for nonconvex stochastic optimization. SIAM Journal on Optimization, 27(2):927–956, 2017.

[34] Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. Spiderboost and momentum: Faster variance reduction algorithms. In Advances in Neural Information Processing Systems, pages 2406–2416, 2019.

[35] Philip Wolfe. Convergence conditions for ascent methods. SIAM Review, 11(2):226–235, 1969.

[36] Philip Wolfe. Convergence conditions for ascent methods II: Some corrections. SIAM Review, 13(2): 185–188, 1971.

[37] Farzad Yousefian, Angelia Nedić, and Uday V Shanbhag. A smoothing stochastic quasi-Newton method for non-Lipschitzian stochastic optimization problems. In 2017 Winter Simulation Conference (WSC), pages 2291–2302. IEEE, 2017.