Fast and Furious Convergence: Stochastic Second-Order Methods under Interpolation

Si Yi Meng[*1] **Sharan Vaswani**[*2] **Issam Laradji**[1,3] **Mark Schmidt**[1,4] **Simon Lacoste-Julien**[2]
[1]University of British Columbia [3]Element AI
[2]Mila, Université de Montréal [4]CCAI Affiliate Chair (Amii)

Abstract

We consider stochastic second-order methods for minimizing smooth and strongly-convex functions under an interpolation condition satisfied by over-parameterized models. Under this condition, we show that the regularized subsampled Newton method (R-SSN) achieves global linear convergence with an adaptive step-size and a constant batch-size. By growing the batch size for both the subsampled gradient and Hessian, we show that R-SSN can converge at a quadratic rate in a local neighbourhood of the solution. We also show that R-SSN attains local linear convergence for the family of self-concordant functions. Furthermore, we analyze stochastic BFGS algorithms in the interpolation setting and prove their global linear convergence. We empirically evaluate stochastic L-BFGS and a "Hessian-free" implementation of R-SSN for binary classification on synthetic, linearly-separable datasets and real datasets under a kernel mapping. Our experimental results demonstrate the fast convergence of these methods, both in terms of the number of iterations and wall-clock time.

1 Introduction

Many machine learning tasks can be formulated as minimizing a finite sum of objective functions. For instance, in supervised learning, each component of the finite sum is the loss function associated with a training example. The overall objective is typically minimized using stochastic first-order methods such as stochastic gradient descent (SGD) or its variance-reduced versions (Schmidt et al., 2017; Johnson and Zhang, 2013; Defazio et al., 2014). However, first-order methods can suffer from slow convergence on ill-conditioned problems. Popular adaptive methods (Duchi et al., 2011; Kingma and Ba, 2015; Tieleman and Hinton, 2012) alleviate this problem to some extent by using the covariance of stochastic gradients to approximate second order information, although there are no worst-case guarantees on the quality of this approximation. Empirically, these adaptive methods are more robust to the problem's conditioning and result in decent performance across different tasks.

There has been substantial work in explicitly incorporating second-order information through the Hessian matrix in Newton methods (Liu and Nocedal, 1989; Nocedal and Wright, 2006; Martens, 2010) or through the Fisher information matrix in natural gradient methods (Amari, 1998; Roux et al., 2008; Pascanu and Bengio, 2014; Martens and Grosse, 2015). Incorporating this information to adapt to the local curvature enables Newton methods to achieve quadratic convergence in a neighbourhood of the solution, whereas typical first-order methods can only achieve linear convergence (Nesterov, 2018). However, the computational complexity of the Newton method is linear in the number of training examples and cubic in the problem dimension, making it impractical for large high-dimensional datasets.

In order to reduce the dependence on the number of training examples, *subsampled Newton* methods (Roosta-Khorasani and Mahoney, 2016a,b; Bollapragada et al., 2018a; Erdogdu and Montanari, 2015; Xu et al., 2016) take a small batch of examples to form the subsampled Hessian in each iteration. In order to further scale these methods to high-dimensional datasets or models with a large number of parameters, there has been a rich body of research in developing computationally efficient approximations to the Hessian matrix (Becker et al., 1988; Liu and Nocedal, 1989; Martens and Grosse, 2015; Erdogdu and Montanari, 2015; Pilanci and Wainwright, 2017; Patel, 2016). In this paper, we mainly restrict our attention to subsam-

[*] Equal contribution.

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

pled Newton methods and consider problems of a moderate dimension where it is computationally efficient to perform subsampled Hessian-vector operations (Werbos, 1988; Pearlmutter, 1994).

Recently, a parallel line of work has focused on optimizing large over-parameterized models prevalent in modern machine learning. The high expressivity of these models enables them to *interpolate* the training data, meaning that the gradient for each component function in the finite sum can become zero at the solution of the overall objective (Schmidt and Roux, 2013; Vaswani et al., 2019a,b; Ma et al., 2018; Bassily et al., 2018). Examples of such models include logistic regression on linearly-separable data, non-parametric regression (Liang and Rakhlin, 2018; Belkin et al., 2019), boosting (Schapire et al., 1998), and over-parameterized deep neural networks (Zhang et al., 2017). Interpolation allows stochastic first-order methods to attain the convergence rates of their deterministic counterparts, without the need for explicit variance reduction (Vaswani et al., 2019a,b; Ma et al., 2018; Bassily et al., 2018). However, first order methods do not take advantage of second order information and therefore, can still be hindered by ill-conditioning. Bertsekas (1997) shows that the Gauss-Newton method is equivalent to Newton's method for solving nonlinear least-squares when interpolation is satisfied; hence the convergence is also superlinear. These reasons motivate us to study stochastic second-order methods in the interpolation setting.

1.1 Contributions

We focus on the regularized subsampled Newton method (R-SSN) that uses a batch of examples to form a subsampled Hessian and gradient vector in each iteration. R-SSN combines the subsampled Hessian with the Levenberg-Marquart (LM) regularization (Levenberg, 1944; Marquardt, 1963) that ensures a unique update direction. We first analyze the convergence rate of R-SSN for strongly-convex functions in the interpolation setting. In Section 3.1, we show that R-SSN with an adaptive step-size and a *constant* batch-size can achieve global linear convergence in expectation. This is in contrast with the work of Bollapragada et al. (2018a) and Bellavia et al. (2018) that analyze subsampled Newton methods without interpolation, and require a *geometrically increasing* batch-size to achieve global linear convergence.

If we allow for a growing batch-size in the interpolation setting, R-SSN results in linear-quadratic convergence in a local neighbourhood of the optimal solution (Section 3.2). In contrast, in order to obtain superlinear convergence, Bollapragada et al. (2018a) require the batch size for the subsampled gradient to grow at a faster-than-geometric rate. Our results thus show that interpolation allows R-SSN to achieve fast convergence with a more reasonable growth of the batch size.

We further analyze the performance of R-SSN for minimizing self-concordant functions. In Section 4, we analyze the convergence of R-SSN for self-concordant functions under the interpolation setting, and show that R-SSN with an adaptive step-size and constant batch-size results in local linear convergence in expectation. Closest to our work is the recent paper by Marteau-Ferey et al. (2019) that shows approximate Newton methods result in local linear convergence independent of the condition number; however, they do not consider the interpolation setting and requires that the approximate Newton directions are "close" to the true direction with high probability.

Next, we study stochastic Quasi-Newton methods (Schraudolph et al., 2007; Sunehag et al., 2009; Zhou et al., 2017; Liu et al., 2018; Berahas et al., 2019; Gao and Goldfarb, 2019) in the interpolation setting (Section 5). We prove that these algorithms, including the popular L-BFGS (Liu and Nocedal, 1989) (the limited-memory version of BFGS), can attain global linear convergence with a constant batch-size. Our result is in contrast to previous work that shows the global linear convergence of stochastic BFGS algorithms by either using variance-reduction (Kolte et al., 2015; Lucchi et al., 2015; Moritz et al., 2016) or progressive batching strategies (Bollapragada et al., 2018b).

Finally, in Section 6, we evaluate R-SSN and stochastic L-BFGS for binary classification. We use synthetic linearly-separable datasets and consider real datasets under a kernel mapping. We use automatic differentiation and truncated conjugate gradient (Hestenes and Stiefel, 1952) to develop a "Hessian-free" implementation of R-SSN that allows computing the Newton update without additional memory overhead.[1] When interpolation holds, we observe that both R-SSN and stochastic L-BFGS result in faster convergence when compared to popular first-order methods (Johnson and Zhang, 2013; Kingma and Ba, 2015; Duchi et al., 2011). Furthermore, a modest batch-growth strategy and stochastic line-search (Vaswani et al., 2019b) scheme ensure that R-SSN is computationally efficient and competitive with stochastic first-order methods and L-BFGS variants in terms of both the number of iterations and wall-clock time required for convergence.

2 Background

We consider unconstrained minimization of a finite sum, $\min_{w \in \mathbb{R}^d} f(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w)$, where f and each f_i are twice continuously differentiable. The regularized

[1] With code: `https://github.com/IssamLaradji/ssn`.

subsampled Newton method (R-SSN) takes a step in the *subsampled Newton direction* at iteration k:

$$w_{k+1} = w_k - \eta_k \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1} \nabla f_{\mathcal{G}_k}(w_k), \qquad (1)$$

where η_k is the step size. The sets $\mathcal{G}_k$ ($b_{g_k} = |\mathcal{G}_k|$) and $\mathcal{S}_k$ ($b_{s_k} = |\mathcal{S}_k|$) are independent samples of indices uniformly chosen from $\{1, \ldots, n\}$ without replacement. The subsampled gradient and the regularized subsampled Hessian are defined as

$$\nabla f_{\mathcal{G}_k}(w_k) = \frac{1}{b_{g_k}} \sum_{i \in \mathcal{G}_k} \nabla f_i(w_k), \qquad (2)$$

$$\mathbf{H}_{\mathcal{S}_k}(w_k) = \frac{1}{b_{s_k}} \sum_{i \in \mathcal{S}_k} \nabla^2 f_i(w_k) + \tau I_d \qquad (3)$$

where $\tau \geq 0$ is the LM regularization (Levenberg, 1944; Marquardt, 1963). Both the subsampled gradient and Hessian are unbiased, implying that $\mathbb{E}_{\mathcal{G}_k}[\nabla f_{\mathcal{G}_k}(w_k)] = \nabla f(w_k)$ and $\mathbb{E}_{\mathcal{S}_k}[\mathbf{H}_{\mathcal{S}_k}(w_k)] = \nabla^2 f(w_k) + \tau I_d$.

Throughout this paper, $\|\cdot\|$ denotes the ℓ_2 norm of a vector or the spectral norm of a matrix. For all our convergence results, we make the standard assumptionsthat f is μ-strongly convex and L-smooth. We denote $\kappa = L/\mu$ as the condition number of f, and w^* as the unique minimizer of f. Additionally, we assume that each component f_i is L_i-smooth and μ_i-strongly-convex. Although μ_i can be zero for some of the components, meaning they are only convex, the finite sum is strongly-convex with $\mu > 0$. Furthermore, we define $\bar{\mu} = \sum_{i=1}^{n} \mu_i / n \leq \mu$ and $\bar{L} = \sum_{i=1}^{n} L_i / n \geq L$ to be the average strong-convexity and smoothness constants of f.

These assumptions imply that for any subsample S, the function $\sum_{i \in S} f_i / |S|$ is $L_{\mathcal{S}}$-smooth and $\mu_{\mathcal{S}}$-strongly-convex, hence the eigenvalues of the corresponding subsampled Hessian can be upper and lower-bounded (Bollapragada et al., 2018a,b; Moritz et al., 2016). In particular, if $\tilde{\mu} = \min_S \mu_S \geq 0$ and $\tilde{L} = \max_S L_S$, then for any sample $\mathcal{S}$ and point w, the *regularized* subsampled Hessian $\mathbf{H}_{\mathcal{S}}(w)$ has eigenvalues bounded in the $[\tilde{\mu} + \tau, \tilde{L} + \tau]$ range. The LM regularization thus ensures that $\mathbf{H}_{\mathcal{S}}(w)$ will always be positive definite, the subsampled Newton direction exists and is unique.

In this work, we focus on models capable of *interpolating* the training data. Typically, these models are highly expressive and are able to fit all the training data. In the finite sum setting, interpolation implies that if $\nabla f(w^*) = 0$, then $\nabla f_i(w^*) = 0$ for all i (Vaswani et al., 2019b,a; Bassily et al., 2018; Ma et al., 2018), meaning that all the individual functions f_i are minimized at the optimal solution w^*. For the smooth, strongly-convex finite sums we consider, interpolation implies the *strong growth condition* (SGC) (Vaswani et al., 2019a, Propositions 1, 2), (Schmidt and Roux, 2013): for some constant $\rho \geq 1$ and all w,

$$\mathbb{E}_i \left\|\nabla f_i(w)\right\|^2 \leq \rho \left\|\nabla f(w)\right\|^2. \qquad \text{(SGC)}$$

For example, if the training data spans the feature space, then the SGC is satisfied for models interpolating the data when using the squared loss (for regression) and the squared hinge loss (for classification).

3 Subsampled Newton Methods

In this section, we present our main theoretical results for R-SSN. We characterize its global linear convergence (Section 3.1) and local quadratic convergence (Section 3.2) under interpolation. We use the notions of Q and R-convergence rates (Ortega and Rheinboldt, 1970; Nocedal and Wright, 2006) reviewed in Appendix G.

3.1 Global convergence

In the following theorem, we show that for smooth and strongly-convex functions satisfying interpolation, R-SSN with an adaptive step-size and constant batch-sizes for both the subsampled gradient and Hessian results in linear convergence from an arbitrary initialization.

Theorem 1 (Global linear convergence)**.** *Under (a) μ-strong-convexity, (b) L-smoothness of f, (c) $[\tilde{\mu} + \tau, \tilde{L} + \tau]$-bounded eigenvalues of the subsampled Hessian and (d) ρ-SGC, the sequence $\{w_k\}_{k \geq 0}$ generated by R-SSN with (i) step-size $\eta_k = \frac{\left(\mu_{\mathcal{S}_k} + \tau\right)^2}{L\left(\left(\mu_{\mathcal{S}_k} + \tau\right) + \left(L_{\mathcal{S}_k} + \tau\right) c_g\right)}$ and (ii) constant batch-sizes $b_{s_k} = b_s$, $b_{g_k} = b_g$ converges to w^* at a Q-linear rate from an arbitrary initialization w_0,*

$$\mathbb{E}[f(w_T)] - f(w^*) \leq (1 - \alpha)^T \left(f(w_0) - f(w^*)\right), \textit{ where}$$

$$\alpha = \min\left\{\frac{(\bar{\mu} + \tau)^2}{2\kappa c_g (\tilde{L} + \tau)}, \frac{(\bar{\mu} + \tau)}{2\kappa(\tilde{L} + \tau)}\right\} \textit{ and } c_g = \frac{(\rho - 1)\,(n - b_g)}{(n - 1)\, b_g}.$$

The proof is given in Appendix A. While the dependence on b_g is explicit, the dependence on b_s is through the constant $\tilde{\mu}$; as b_s tends to n, $\mu_{\mathcal{S}_k}$ tends to μ, allowing R-SSN to use a larger step-size that results in faster convergence (since $\mu \geq \bar{\mu}$). If we set $b_g = b_s = n$ and $\tau = 0$, we obtain the rate $\mathbb{E}[f(w_T)] - f(w^*) \leq (1 - 1/(2\kappa^2))^T \left[\mathbb{E}[f(w_0)] - f(w^*)\right]$ which matches the deterministic rate (Karimireddy et al., 2018, Theorem 2) up to a constant factor of 2. The minimum batch-size required to achieve the deterministic rate is $b_g \geq n/(1 + (\bar{\mu} + \tau)(n-1)/(\rho - 1))$. Similar to SGD (Vaswani et al., 2019a; Schmidt and Roux, 2013), the interpolation condition allows R-SSN with a constant batch-size to obtain Q-linear convergence. In the absence of interpolation, SSN has only been shown to achieve an R-linear rate by increasing the batch size geometrically for the subsampled gradient (Bollapragada et al., 2018a). Next, we analyze the convergence properties of R-SSN in a local neighbourhood of the solution.

3.2 Local convergence

To analyze the local convergence of R-SSN, we make additional assumptions. Similar to other local convergence results (Roosta-Khorasani and Mahoney, 2016b), we assume that the Hessian is M-Lipschitz continuous, implying that for all $w, v \in \mathbb{R}^d$, $\left\|\nabla^2 f(w) - \nabla^2 f(v)\right\| \leq M \left\|w - v\right\|$. We also assume that the *moments of the iterates are bounded*, implying that for all iterates w_k, there exists a constant γ such that $\mathbb{E}\left[\left\|w_k - w^*\right\|^2\right] \leq \gamma \left[\mathbb{E}\left\|w_k - w^*\right\|\right]^2$ where the expectation is taken over the entire history up until step k. If the iterates lie within a bounded set, then the above assumption holds for some finite γ (Bollapragada et al., 2018a; Berahas et al., 2020; Harikandeh et al., 2015). We also assume that the sample variance of the subsampled Hessian is bounded, namely $\frac{1}{n-1}\sum_{i=1}^n\left[\left\|\nabla^2 f_i(w) - \nabla^2 f(w)\right\|^2\right] \leq \sigma^2$ for some $\sigma > 0$.

Theorem 2 (Local convergence)**.** *Under the same assumptions (a) - (d) of Theorem 1 along with (e) M-Lipschitz continuity of the Hessian, (f) γ-bounded moments of iterates, and (g) σ-bounded variance of the subsampled Hessian, the sequence $\{w_k\}_{k\geq 0}$ generated by R-SSN with (i) unit step-size $\eta_k = 1$ and (ii) growing batch-sizes satisfying*

$$b_{g_k} \geq \frac{n}{\left(\frac{n-1}{\rho-1}\right)\left\|\nabla f(w_k)\right\|^2 + 1},\; b_{s_k} \geq \frac{n}{\frac{n}{\sigma^2}\left\|\nabla f(w_k)\right\| + 1}$$

converges to w^* at a linear-quadratic rate

$$\begin{aligned}\mathbb{E}\left\|w_{k+1} - w^*\right\| \leq & \frac{\gamma(M + 2L + 2L^2)}{2(\tilde{\mu} + \tau)}\left(\mathbb{E}\left\|w_k - w^*\right\|\right)^2 \\ & + \frac{\tau}{\tilde{\mu} + \tau}\mathbb{E}\left\|w_k - w^*\right\|\end{aligned}$$

in a local neighbourhood of the solution $\left\|w_0 - w^*\right\| \leq \frac{2(\tilde{\mu}+\tau)}{\gamma(M+2L+2L^2)}$. Furthermore, when $\tau = 0$ and $\tilde{\mu} > 0$, R-SSN can achieve local quadratic convergence

$$\mathbb{E}\left\|w_{k+1} - w^*\right\| \leq \gamma\left(\frac{M + 2L + 2L^2}{2\tilde{\mu}}\right)\left(\mathbb{E}\left\|w_k - w^*\right\|\right)^2.$$

The proof is provided in Appendix B. Note that the constant $\frac{\tau}{\tilde{\mu}+\tau}$ on the linear term is less than one and hence contraction is guaranteed, while the constant on the quadratic term is not required to be less than one to guarantee convergence (Nesterov, 2018).

This theorem states that if we progressively increase the batch size for both the subsampled Hessian and the subsampled gradient, then we can obtain a local linear-quadratic convergence rate. For inexact Newton methods to obtain quadratic convergence, it has been shown that the error term needs to decrease as a quadratic function of the gradient (Dembo et al., 1982), which is not provided by SGC, and hence the need for additional techniques such as batch growing. The required geometric growth rate for $\mathcal{G}_k$ is the same as that of SGD to obtain linear convergence without variance-reduction or interpolation (De et al., 2016; Friedlander and Schmidt, 2012). Note that the proof can be easily modified to obtain a slightly worse linear-quadratic rate when using a subsampled Hessian with a constant batch-size. In addition, following Ye et al. (2017), we can relax the Lipschitz Hessian assumption and obtain a slower superlinear convergence. Unlike the explicit quadratic rate above, in the absence of interpolation, SSN without regularization has only been shown to achieve an asymptotic superlinear rate (Bollapragada et al., 2018a). Moreover, in this case, b_{g_k} needs to increase at a rate that is faster than geometric, considerably restricting its practical applicability.

The following corollary (proved in Appendix B.1) states that if we decay the LM-regularization sequence τ_k proportional to the gradient norm, R-SSN can achieve quadratic convergence for strongly-convex functions. In fact, this decay rate is inversely proportional to the growth of the batch size for the subsampled Hessian, indicating that larger batch-sizes require smaller regularization. This relationship between the regularization and sample size is consistent with the observations of Ye et al. (2017).

Corollary 1. *Under the same assumptions as Theorem 2, if we decrease the regularization term according to $\tau_k \leq \left\|\nabla f(w_k)\right\|$, R-SSN can achieve local quadratic convergence with $\left\|w_0 - w^*\right\| \leq \frac{2(\tilde{\mu}+\tau_k)}{\gamma(M+4L+2L^2)}$:*

$$\mathbb{E}\left\|w_{k+1} - w^*\right\| \leq \frac{\gamma(M + 4L + 2L^2)}{2(\tilde{\mu}_k + \tau_k)}\left(\mathbb{E}\left\|w_k - w^*\right\|\right)^2,$$

where $\tilde{\mu}_k$ is the minimum eigenvalue of $\nabla^2 f_{\mathcal{S}_k}(w_k)$ over all batches of size $|\mathcal{S}_k|$.

Since $\tilde{\mu} \leq \tilde{\mu}_k$, this gives a tighter guarantee on the decrease in suboptimality at every iteration as the batch size grows and τ_k decreases. On the other hand, if we make a stronger growth assumption on the stochastic gradients such that $\mathbb{E}_i\left\|\nabla f_i(w)\right\|^2 \leq \rho\left\|\nabla f(w)\right\|^4$, then R-SSN can achieve local quadratic convergence using only a *constant* batch-size for the subsampled gradient and the same growth rate for the subsampled Hessian. We state this result as Corollary 2, but we acknowledge that this assumption might be too restrictive to be useful in practice.

4 R-SSN under Self-Concordance

We now analyze the convergence of R-SSN for the class of strictly-convex self-concordant functions. A function

f is called *standard self-concordant*[2] if the following inequality holds for any w and direction $u \in \mathbb{R}^d$,

$$|u^\intercal \left(f'''(w)[u]\right) u| \leq 2 \, \|u\|^{3/2}_{\nabla^2 f(w)} \ .$$

Here, $f'''(w)$ denotes the third derivative tensor at w and $f'''(w)[u]$ represents the matrix corresponding to the projection of this tensor on vector u. Functions in this (generalized) class include common losses such as the Huber loss (Marteau-Ferey et al., 2019), smoothed hinge loss (Zhang and Lin, 2015), and the logistic loss (Bach et al., 2010). Self-concordant functions enable an affine-invariant analysis of Newton's method and have convergence rates similar to that of smooth and strongly-convex functions, but without problem-dependent constants (Nesterov, 2018; Boyd and Vandenberghe, 2004). To characterize the convergence rate of R-SSN under self-concordance, we define the regularized Newton decrement at w as:

$$\lambda(w) := \langle \nabla f(w) \, , \, \left[\nabla^2 f(w) + \tau I_d\right]^{-1} \nabla f(w) \rangle^{1/2}$$

When $\tau = 0$, this corresponds to the standard definition of the Newton decrement (Nesterov, 2018). If we denote the standard Newton decrement as $\lambda^0(w)$, then the regularized version can be bounded as $\lambda(w) \leq \lambda^0(w)$. We also introduce the regularized stochastic Newton decrement as

$$\tilde{\lambda}_{i,j}(w) := \langle \nabla f_i(w) \, , \, [\mathbf{H}_j(w)]^{-1} \nabla f_i(w) \rangle^{1/2}$$

for independent samples i and j. In Proposition 1 (Appendix E), we show that if f satisfies the SGC and the subsampled Hessian has bounded eigenvalues, then f satisfies a similar growth condition for the regularized stochastic Newton decrement: for all w and j, there exists $\rho_{\text{nd}} \geq 1$ such that

$$\mathbb{E}_i \left[\tilde{\lambda}^2_{i,j}(w)\right] \leq \rho_{\text{nd}} \, \lambda^2(w). \quad \text{(Newton decrement SGC)}$$

For ease of notation, we use $\tilde{\lambda}_k$ to denote the regularized stochastic Newton decrement computed using samples $\mathcal{G}_k$ and $\mathcal{S}_k$. We now consider an update step similar to (1) but with a step-size adjusted to $\tilde{\lambda}_k$,

$$w_{k+1} = w_k - \frac{c \, \eta}{1 + \eta \, \tilde{\lambda}_k} \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1} \nabla f_{\mathcal{G}_k}(w_k) \qquad (4)$$

where $c, \eta \in (0, 1]$. For the next theorem, we assume the iterates lie in a bounded set such that $\|w_k - w^*\| \leq D$ for all k (Harikandeh et al., 2015).

Theorem 3. *Under (a) self-concordance (b) L-smoothness of f (c) $[\tilde{\mu} + \tau, \tilde{L} + \tau]$-bounded eigenvalues of the subsampled Hessian (d) $\rho_{nd} = \frac{\rho L}{\tilde{\mu}+\tau}$-Newton decrement SGC and (e) bounded iterates, the sequence $\{w_k\}_{k \in [0,m]}$ generated by R-SSN in (4) with (i) $c = \sqrt{\frac{\tilde{\mu}+\tau}{L}}$, $\eta \in \left(0, \frac{c}{\rho_{nd}(1+\tilde{L}D/(\tilde{\mu}+\tau))}\right]$ and (ii) constant batch sizes ≥ 1 converges to w^* from an arbitrary initialization w_0 at a rate characterized by*

$$\mathbb{E}\left[f(w_{k+1})\right] \leq f(w_k) - \eta \, \delta \, \omega\left(\lambda_k\right).$$

Here $\delta \in (0, 1]$ and the univariate function ω is defined as $\omega(t) = t - \ln(1 + t)$. Furthermore, in the local neighbourhood where $\lambda_m \leq 1/6$, the sequence $\{w_k\}_{k \geq m}$ converges to w^* at a Q-linear rate,

$$\mathbb{E}\left[f(w_T)\right] - f(w^*) \leq \left(1 - \frac{\eta\delta}{1.26}\right)^{T-m} \left(\mathbb{E}\left[f(w_m)\right] - f(w^*)\right).$$

The proof is given in Appendix C. The above result gives an insight into the convergence properties of R-SSN for loss functions that are self-concordant but not necessarily strongly-convex. Note that the analysis of the deterministic Newton's method for self-concordant functions yields a problem-independent local quadratic convergence rate, meaning it does not rely on the condition number of f. However, it is non-trivial to improve the above analysis and obtain a similar affine-invariant result. Similarly, the algorithm parameters for the inexact Newton method in Zhang and Lin (2015) depend on the condition number.

5 Stochastic BFGS

Consider the stochastic BFGS update,

$$w_{k+1} = w_k - \eta_k \, \mathbf{B}_k \, \nabla f_{\mathcal{G}_k}(w_k) \qquad (5)$$

where $\mathbf{B}_k$ is a positive definite matrix constructed to approximate the inverse Hessian $\left[\nabla^2 f(w_k)\right]^{-1}$. As in previous works (Bollapragada et al., 2018b; Moritz et al., 2016; Lucchi et al., 2015; Sunehag et al., 2009), we assume that $\mathbf{B}_k$ has bounded eigenvalues such that $\lambda_1 I \preceq \mathbf{B}_k \preceq \lambda_d I$. We now show that under SGC, stochastic BFGS achieves global linear convergence with a constant step-size.

Theorem 4. *Under (a) μ-strong-convexity, (b) L-smoothness of f, (c) $[\lambda_1, \lambda_d]$-bounded eigenvalues of $\mathbf{B}_k$ and (d) ρ-SGC, the sequence $\{w_k\}_{k \geq 0}$ generated by stochastic BFGS with (i) a constant step-size $\eta_k = \eta = \frac{\lambda_1}{c_g L \lambda_d^2}$ and (ii) and a constant batch size $b_{g_k} = b_g$ converges to w^* at a linear rate from an arbitrary initialization w_0,*

$$\mathbb{E}[f(w_T)] - f(w^*) \leq \left(1 - \frac{\mu \lambda_1^2}{c_g \, L \lambda_d^2}\right)^T \left(f(w_0) - f(w^*)\right)$$

where $c_g = \frac{(n-b_g)\,(\rho-1)}{(n-1)\,b_g} + 1$.

[2] If f is self-concordant with parameter $C \geq 0$, re-scaling it by a factor $C^2/4$ makes it standard self-concordant (Nesterov, 2018, Corollary 5.1.3), and hence we consider only standard self-concordant functions in this work.

The proof is given in Appendix D. The strong-convexity assumption can be relaxed to the Polyak-Łojasiewicz inequality (Polyak, 1963; Karimi et al., 2016) while giving the same rate up to a constant factor. In the absence of interpolation, global linear convergence can only be obtained by using either variance-reduction techniques (Moritz et al., 2016; Lucchi et al., 2015) or progressive batching (Bollapragada et al., 2018b). Similar to these works, our analysis for the stochastic BFGS method applies to preconditioned SGD where $\mathbf{B}_k$ can be any positive-definite preconditioner.

6 Experiments

We verify our theoretical results on a binary classification task on both synthetic and real datasets. We evaluate two variants of R-SSN: `R-SSN-const` that uses a constant batch-size and `R-SSN-grow` where we grow the batch size geometrically (by a constant multiplicative factor of 1.01 in every iteration (Friedlander and Schmidt, 2012)). Note that although our theoretical analysis of R-SSN requires independent batches for the subsampled gradient and Hessian, we use the same batch for both variants of R-SSN and observe that this does not adversely affect empirical performance. We use truncated conjugate gradient (CG) (Hestenes and Stiefel, 1952) to solve for the (subsampled) Newton direction in every iteration. For each experiment, we choose the LM regularization τ via a grid search. For `R-SSN-grow`, following Corollary 1, starting from the LM regularization selected by grid search, we progressively decrease τ in the same way as we increase the batch size. We evaluate stochastic L-BFGS (`sLBFGS`) with a "memory" of 10. For `sLBFGS`, we use the same minibatch to compute the difference in the (subsampled) gradients to be used in the inverse Hessian approximation (this corresponds to the "full" overlap setting in Bollapragada et al. (2018b)), to which we add a small regularization to ensure positive-definiteness. For both R-SSN and `sLBFGS`, we use the stochastic line-search from Vaswani et al. (2019b) to set the step size.

We compare the proposed algorithms against common first-order methods: SGD, SVRG (Johnson and Zhang, 2013), Adam (Kingma and Ba, 2015) and Adagrad (Duchi et al., 2011). For all experiments, we use an (initial) batch size of $b = 100$ and run each algorithm for 200 epochs. Here, an epoch is defined as one full pass over the dataset and does not include additional function evaluations from the line-search or CG. Subsequently, in Figure 2, we plot the mean wall-clock time per epoch that takes these additional computations into account. For SGD, we compare against its generic version and with Polyak acceleration (Polyak, 1964), where in both cases the step size is chosen via stochastic line-search (Vaswani et al., 2019b) with the same hyperparameters, and the acceleration hyperparameter is chosen via grid search. For SVRG, the step size is chosen via 3-fold cross validation on the training set, and we set the number of inner iterations per outer full-gradient evaluation to n/b. We use the default hyperparameters for Adam and Adagrad for their adaptive properties. All results are averaged across 5 runs.

Synthetic datasets: We first evaluate the algorithms on binary classification using synthetic and linearly-separable datasets with varying margins. Linear separability ensures the interpolation condition will hold. For each margin, we generate a dataset with 10k examples with $d = 20$ features and binary labels. For these datasets, we also compare against unregularized Newton and L-BFGS, both using the full-batch (hence deterministic). For L-BFGS, we use the PyTorch (Paszke et al., 2017) implementation with line search and an initial step size of 0.9.

In Figure 1, we show the training loss for the logistic loss (row 1) and the squared hinge loss (row 2). We observe that by incorporating second order information, R-SSN can converge much faster than first order methods. In addition, global linear convergence can be obtained using only constant batch-sizes, verifying Theorem 1. Furthermore, by growing the batch size, R-SSN performs similar to the deterministic Newton method, verifying the local quadratic convergence of Theorem 2. Although our theory only guarantees linear convergence globally (instead of superlinear) for stochastic L-BFGS under interpolation, these methods can be much faster than first order methods empirically. Finally, as we increase the margin (left to right of Figure 1), the theoretical rate under interpolation improves (Vaswani et al., 2019a), resulting in faster and more stable convergence for the proposed algorithms.

Real datasets: We also consider real datasets `mushrooms`, `rcv1`, and `ijcnn` from the LIBSVM repository (Chang and Lin, 2011), and use an $80 : 20$ split for the training and test set, respectively. We fit a linear model under the radial basis function (RBF) kernel. The kernel mapping results in the effective dimension being equal to the number of points in the dataset. This results in problem dimensions of 6.5k, 20k and 28k for `mushrooms`, `rcv1`, and `ijcnn` respectively. The RBF-kernel bandwidths are chosen via grid search using 10-fold cross validation on the training split following Vaswani et al. (2019b). Note that the `mushrooms` dataset is linearly separable under the chosen kernel mapping and thus satisfies the interpolation condition. For these datasets, we limit the maximum batch size to 8192 for `SSN-grow` to alleviate the computation and memory overhead. We show the training loss, test accuracy, as well as the mean wall-clock time

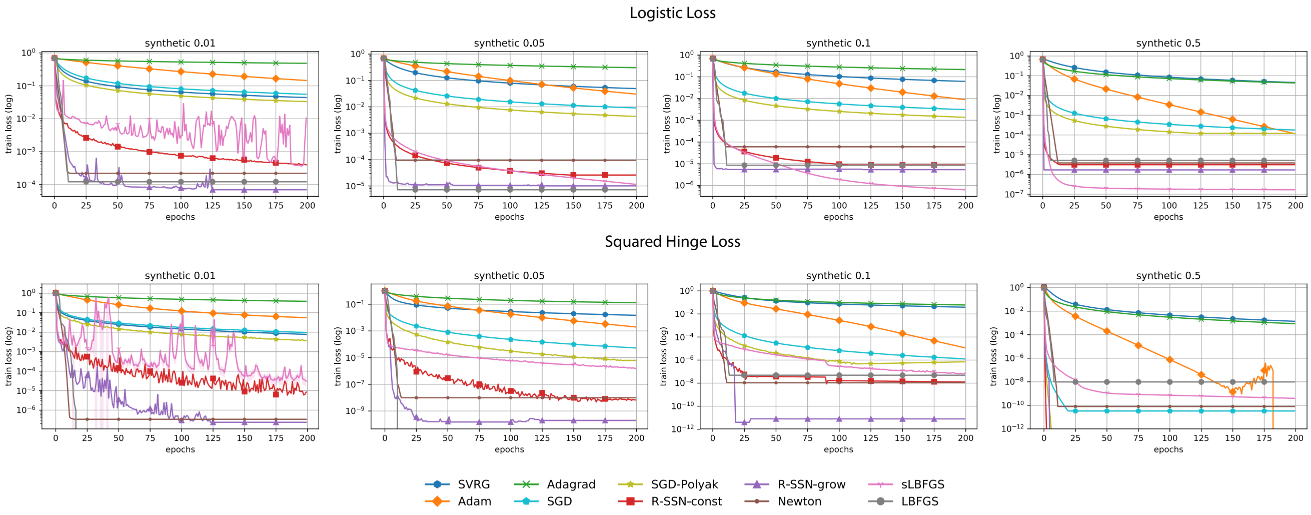

Figure 1: Comparison of R-SSN variants and stochastic L-BFGS against first order methods on synthetic data where interpolation is satisfied. Both R-SSN and stochastic L-BFGS outperform first order methods. From left to right we show the results for datasets with margins `[0.01, 0.05, 0.1, 0.5]` under two different losses.

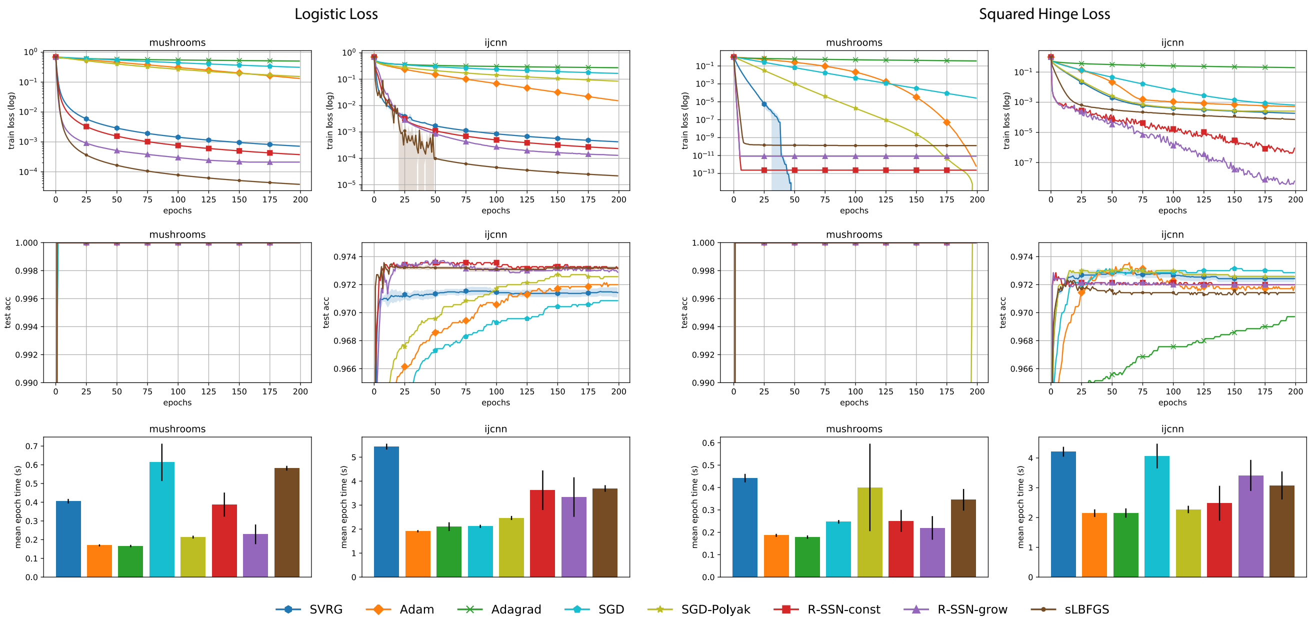

Figure 2: Comparison of R-SSN variants and stochastic L-BFGS against first order methods on real datasets. Under the RBF kernel mapping, the `mushrooms` dataset satisfies interpolation, and the R-SSN variants and `sLBFGS` perform best in this setting. When interpolation is not satisfied (`ijcnn`), these methods are still competitive.

per epoch. For this set of experiments, the stochastic line-search procedure (Vaswani et al., 2019b) used in conjunction with `sLBFGS` can lead to a large number of backtracking iterations, resulting in a high wall-clock time per epoch. To overcome this issue, we use a constant step-size variant of `sLBFGS` and perform a grid search over $[10^{-4}, 1]$ to select the best step-size for each experiment. In the interest of space, we refer the reader to Appendix F for results on the `rcv1` dataset.

In the first row of Figure 2, we observe when interpolation is satisfied (`mushrooms`), the R-SSN variants and `sLBFGS` outperform all other methods in terms of training loss convergence. When interpolation is not satisfied (`ijcnn`), the stochastic second-order methods are still competitive with the best performing method. In the second row, we observe that despite the fast convergence of these methods, generalization performance does not deteriorate regardless of whether interpolation is satisfied. Furthermore, since all our experiments are run on the GPU, we take advantage of parallelization and amortize the runtime of the proposed methods. This is reflected in the third row, where we plot the mean per-epoch wall-clock time for all methods. Also note that `sLBFGS` is competitive with R-SSN in terms of the number of iterations, but has a higher per-epoch cost on average.

7 Related Work

Besides the works discussed earlier, there have been many recent works analyzing subsampled Newton methods (Byrd et al., 2011, 2012; Milzarek et al., 2019; Li et al., 2019). In particular, Roosta-Khorasani and Mahoney (2016a,b) derive probabilistic global and local convergence rates. Similarly, Xu et al. (2016) achieve a local linear-quadratic convergence in high probability by using non-uniform sampling to select the batch. Erdogdu and Montanari (2015) propose the NewSamp method that subsamples the Hessian followed by a regularized truncated SVD to form the inverse Hessian approximation. Ye et al. (2017) provide a unified analysis of the local convergence properties of well-known variants of subsampled Newton's method where the gradient is assumed to be exact. For nonconvex objectives, Bergou et al. (2018) show second order results for the global convergence of subsampled Newton's method with line search.

Recently, stochastic optimization under an interpolation condition has been analyzed by several authors. Both Ma et al. (2018) and Vaswani et al. (2019a) show that under mild conditions, SGD with constant step-size achieves linear convergence for strongly-convex functions. Additionally, it can match the deterministic rates in the convex (Vaswani et al., 2019a; Schmidt and Roux, 2013; Cevher and Vũ, 2019) and non-convex (Bassily et al., 2018; Vaswani et al., 2019a) settings. The interpolation assumption also allows for momentum-type methods to achieve the accelerated rates of convergence for least-squares (Gower et al., 2018) and more generally in convex settings (Vaswani et al., 2019a; Liu and Belkin, 2016). Although the step size in these settings depends on unknown quantities, it has been recently shown that stochastic line-search methods based on the Armijo condition can be used to automatically set the step size and still achieve fast convergence rates (Vaswani et al., 2019b). In contrast to previous works on stochastic second-order methods, we consider fully stochastic R-SSN and show global Q-linear convergence and local Q-quadratic convergence in expectation for SSN for strongly-convex objectives under the interpolation condition.

The class of self-concordant functions was first introduced by Nesterov and Nemirovskii (1994) and has been extended to include common losses such as (multi-class) logistic regression (Bach et al., 2010). Zhang and Lin (2015) propose a distributed inexact Newton's method and provide convergence analysis for self-concordant functions. Mokhtari et al. (2016); Eisen et al. (2018) introduce the Ada Newton method that computes a geometrically increasing batch-size based on a targeted statistical accuracy and show local quadratic convergence for self-concordant functions with high probability. However, none of these works analyze stochastic Newton methods for self-concordant functions.

Quasi-Newton methods that approximate the Hessian are popular in practice due to their cheaper iteration costs compared to full Newton methods while having superlinear convergence (Dennis and Moré, 1974). Stochastic variants of these methods have been proposed to further reduce the computation overhead from a large training set (Schraudolph et al., 2007; Zhou et al., 2017; Liu et al., 2018; Berahas et al., 2019; Gao and Goldfarb, 2019). Combined with variance-reduction techniques from popular first-order methods, stochastic BFGS-type methods can achieve global linear convergence (Kolte et al., 2015; Lucchi et al., 2015; Moritz et al., 2016). Bollapragada et al. (2018b) use progressive batching with stochastic L-BFGS to show linear convergence for strongly-convex functions and sublinear rate for convex functions. Mokhtari et al. (2018) use memory to reduce the variance for stochastic BFGS and achieve a local superlinear convergence rate. Kelley (2002)'s implicit filtering method can achieve superlinear convergence using a deterministic and finite-difference-based BFGS for minimizing objectives whose noise decreases as we approach the global minimum, similar to the interpolation setting. Our work fills in the gap between stochastic quasi-Newton methods and the interpolation condition.

8 Conclusion

We showed that the regularized subsampled Newton method with a constant batch-size achieves linear convergence rates when minimizing smooth functions that are strongly-convex or self-concordant under the interpolation setting. We also showed that interpolation enables stochastic BFGS-type methods to converge linearly. We validated our theoretical claims via experiments using kernel functions and demonstrated the fast convergence of the proposed methods. Our theoretical and empirical results show the potential for training large over-parameterized models that satisfy the interpolation property. For future work, we aim to investigate ways to handle non-convex losses and further scale these methods to optimize over millions of parameters.

9 Acknowledgements

We would like to thank Raghu Bollapragada, Nicolas Le Roux, Frederik Kunstner, Chris Liaw and Aaron Mishkin for helpful discussions. This research was partially supported by the Canada CIFAR AI Chair Program, by the NSERC Discovery Grants RGPIN-2017-06936 and 2015-06068, an IVADO postdoctoral scholarship, and an NSERC Canada Graduate Scholarships-Master's. Simon Lacoste-Julien is a CIFAR Associate Fellow in the Learning in Machines & Brains program and a Canada CIFAR AI Chair.

References

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

Francis Bach et al. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4:384–414, 2010.

Raef Bassily, Mikhail Belkin, and Siyuan Ma. On exponential convergence of SGD in non-convex over-parametrized learning. *arXiv preprint arXiv:1811.02564*, 2018.

Sue Becker, Yann Le Cun, et al. Improving the convergence of back-propagation learning with second order methods. In *Proceedings of the 1988 connectionist models summer school*, 1988.

Mikhail Belkin, Alexander Rakhlin, and Alexandre B. Tsybakov. Does data interpolation contradict statistical optimality? In *AISTATS*, 2019.

Stefania Bellavia, Natasa Krejic, and Natasa Krklec Jerinkic. Subsampled Inexact Newton methods for minimizing large sums of convex functions. *arXiv preprint arXiv:1811.05730*, 2018.

Albert S Berahas, Majid Jahani, and Martin Takáč. Quasi-newton methods for deep learning: Forget the past, just sample. *arXiv preprint arXiv:1901.09997*, 2019.

Albert S Berahas, Raghu Bollapragada, and Jorge Nocedal. An investigation of Newton-sketch and subsampled Newton methods. *Optimization Methods and Software*, pages 1–20, 2020.

El-houcine Bergou, Youssef Diouane, Vyacheslav Kungurtsev, and Clément W Royer. A subsampling line-search method with second-order results. *arXiv preprint arXiv:1810.07211*, 2018.

Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 1997.

Raghu Bollapragada, Richard H Byrd, and Jorge Nocedal. Exact and inexact subsampled Newton methods for optimization. *IMA Journal of Numerical Analysis*, 39(2):545–578, 2018a.

Raghu Bollapragada, Dheevatsa Mudigere, Jorge Nocedal, Hao-Jun Michael Shi, and Ping Tak Peter Tang. A progressive batching L-BFGS method for machine learning. In *ICML*, 2018b.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Richard H Byrd, Gillian M Chin, Will Neveitt, and Jorge Nocedal. On the use of stochastic Hessian information in optimization methods for machine learning. *SIAM Journal on Optimization*, 21(3):977–995, 2011.

Richard H Byrd, Gillian M Chin, Jorge Nocedal, and Yuchen Wu. Sample size selection in optimization methods for machine learning. *Mathematical programming*, 134(1):127–155, 2012.

Eric Carlen. Trace inequalities and quantum entropy: an introductory course. *Entropy and the quantum*, 529:73–140, 2010.

Volkan Cevher and Bang Công Vũ. On the linear convergence of the stochastic gradient method with constant step-size. *Optimization Letters*, 13(5):1177–1187, 2019.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

Soham De, Abhay Yadav, David Jacobs, and Tom Goldstein. Big batch SGD: Automated inference using adaptive batch sizes. *arXiv preprint arXiv:1610.05792*, 2016.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, 2014.

Ron S Dembo, Stanley C Eisenstat, and Trond Steihaug. Inexact newton methods. *SIAM Journal on Numerical analysis*, 19:400–408, 1982.

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.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Mark Eisen, Aryan Mokhtari, and Alejandro Ribeiro. Large Scale Empirical Risk Minimization via Truncated Adaptive Newton Method. In *AISTATS*, 2018.

Murat A. Erdogdu and Andrea Montanari. Convergence rates of sub-sampled Newton methods. In *NIPS*, 2015.

Michael P Friedlander and Mark Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, 2012.

Wenbo Gao and Donald Goldfarb. Quasi-Newton methods: superlinear convergence without line searches for self-concordant functions. *Optimization Methods and Software*, 34(1):194–217, 2019.

Robert Gower, Filip Hanzely, Peter Richtárik, and Sebastian U Stich. Accelerated stochastic matrix inversion: general theory and speeding up BFGS rules for faster second-order optimization. In *NeurIPS*, 2018.

Reza Harikandeh, Mohamed Osama Ahmed, Alim Virani, Mark Schmidt, Jakub Konečnỳ, and Scott Sallinen. Stop wasting my gradients: Practical SVRG. In *NIPS*, 2015.

Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*. NBS Washington, DC, 1952.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, 2013.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *ECML*, 2016.

Sai Praneeth Karimireddy, Sebastian U Stich, and Martin Jaggi. Global linear convergence of Newton's method without strong-convexity or Lipschitz gradients. *arXiv preprint arXiv:1806.00413*, 2018.

Carl Tim Kelley. A brief introduction to implicit filtering. Technical Report CRSC-TR02-28, North Carolina State University. Center for Research in Scientific Computation, 2002. URL https://repository.lib.ncsu.edu/handle/1840.4/550.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

Ritesh Kolte, Murat Erdogdu, and Ayfer Ozgur. Accelerating SVRG via second-order information. In *NIPS Workshop on Optimization for Machine Learning*, 2015.

Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

Xiang Li, Shusen Wang, and Zhihua Zhang. Do subsampled newton methods work for high-dimensional data? *arXiv preprint arXiv:1902.04952*, 2019.

Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel" ridgeless" regression can generalize. *arXiv preprint arXiv:1808.00387*, 2018.

Chaoyue Liu and Mikhail Belkin. Clustering with Bregman Divergences: an Asymptotic Analysis. In *NIPS*, 2016.

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

Jie Liu, Yu Rong, Martin Takác, and Junzhou Huang. On the acceleration of l-bfgs with second-order information and stochastic batches. *arXiv preprint arXiv:1807.05328*, 2018.

Sharon L Lohr. *Sampling: Design and Analysis: Design and Analysis*. Chapman and Hall/CRC, 2019.

Aurelien Lucchi, Brian McWilliams, and Thomas Hofmann. A variance reduced stochastic Newton method. *arXiv preprint arXiv:1503.08316*, 2015.

Siyuan Ma, Raef Bassily, and Mikhail Belkin. The Power of Interpolation: Understanding the Effectiveness of SGD in Modern Over-parametrized Learning. In *ICML*, 2018.

Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11 (2):431–441, 1963.

Ulysse Marteau-Ferey, Francis Bach, and Alessandro Rudi. Globally Convergent Newton Methods for Ill-conditioned Generalized Self-concordant Losses. In *NeurIPS*, 2019.

Ulysse Marteau-Ferey, Dmitrii Ostrovskii, Francis Bach, and Alessandro Rudi. Beyond Least-Squares: Fast Rates for Regularized Empirical Risk Minimization through Self-Concordance. In *COLT*, 2019.

James Martens. Deep learning via Hessian-free optimization. In *ICML*, 2010.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *ICML*, 2015.

Andre Milzarek, Xiantao Xiao, Shicong Cen, Zaiwen Wen, and Michael Ulbrich. A stochastic semismooth Newton method for nonsmooth nonconvex optimization. *SIAM Journal on Optimization*, 29(4):2916–2948, 2019.

Aryan Mokhtari, Hadi Daneshmand, Aurélien Lucchi, Thomas Hofmann, and Alejandro Ribeiro. Adaptive Newton Method for Empirical Risk Minimization to Statistical Accuracy. In *NIPS*, 2016.

Aryan Mokhtari, Mark Eisen, and Alejandro Ribeiro. IQN: An incremental quasi-Newton method with local superlinear convergence rate. *SIAM Journal on Optimization*, 28(2):1670–1698, 2018.

Philipp Moritz, Robert Nishihara, and Michael Jordan. A linearly-convergent stochastic L-BFGS algorithm.

In *Artificial Intelligence and Statistics*, pages 249–258, 2016.

Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13. Siam, 1994.

Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30. Siam, 1970.

Razvan Pascanu and Yoshua Bengio. Revisiting natural gradient for deep networks. In *ICLR*, 2014.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS*, 2017.

Vivak Patel. Kalman-based stochastic gradient method with stop condition and insensitivity to conditioning. *SIAM Journal on Optimization*, 26(4):2620–2648, 2016.

Barak A. Pearlmutter. Fast Exact Multiplication by the Hessian. *Neural Computation*, 6(1):147–160, 1994.

Mert Pilanci and Martin J Wainwright. Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM Journal on Optimization*, 27(1):205–245, 2017.

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

Boris Teodorovich Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 1963.

Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled newton methods i: globally convergent algorithms. *arXiv preprint arXiv:1601.04737*, 2016a.

Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled newton methods ii: Local convergence rates. *arXiv preprint arXiv:1601.04738*, 2016b.

Nicolas L Roux, Pierre-Antoine Manzagol, and Yoshua Bengio. Topmoumoute online natural gradient algorithm. In *NIPS*, 2008.

Robert E Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

Mark Schmidt and Nicolas Le Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 2017.

Nicol N Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-Newton method for online convex optimization. In *AISTATS*, 2007.

Peter Sunehag, Jochen Trumpf, SVN Vishwanathan, and Nicol Schraudolph. Variable metric stochastic approximation theory. In *AISTATS*, 2009.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 2012.

Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 2015.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and Faster Convergence of SGD for Over-Parameterized Models and an Accelerated Perceptron. In *AISTATS*, 2019a.

Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. In *NeurIPS*, 2019b.

PJ Werbos. Backpropagation: Past and future. In *Proceedings of the Second International Conference on Neural Network*. IEEE, 1988.

Peng Xu, Jiyan Yang, Farbod Roosta-Khorasani, Christopher Ré, and Michael W Mahoney. Sub-sampled Newton methods with non-uniform sampling. In *NIPS*, 2016.

Haishan Ye, Luo Luo, and Zhihua Zhang. Approximate Newton Methods and Their Local Convergence. In *ICML*, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Yuchen Zhang and Xiao Lin. Disco: Distributed optimization for self-concordant empirical loss. In *ICML*, 2015.

Chaoxu Zhou, Wenbo Gao, and Donald Goldfarb. Stochastic Adaptive Quasi-Newton Methods for Minimizing Expected Values. In *ICML*, 2017.