A Fully Stochastic Second-Order Trust Region Method

Frank E. Curtis and Rui Shi

Department of Industrial and Systems Engineering, Lehigh University

COR@L Technical Report 19T-025

A Fully Stochastic Second-Order Trust Region Method

Frank E. Curtis[*1] and Rui Shi[†1]

[1]Department of Industrial and Systems Engineering, Lehigh University

November 15, 2019

Abstract

A stochastic second-order trust region method is proposed, which can be viewed as a second-order extension of the *trust-region-ish* (TRish) algorithm proposed by Curtis et al. [INFORMS J. Optim. 1(3) 200–220, 2019]. In each iteration, a search direction is computed by (approximately) solving a trust region subproblem defined by stochastic gradient and Hessian estimates. The algorithm has convergence guarantees for stochastic minimization in the fully stochastic regime, meaning that guarantees hold when each stochastic gradient is required merely to be an unbiased estimate of the true gradient with bounded variance and when the stochastic Hessian estimates are bounded uniformly in norm. The algorithm is also equipped with a worst-case complexity guarantee in the nearly deterministic regime, i.e., when the stochastic gradient and Hessian estimates are very close in expectation to the true gradients and Hessians. The results of numerical experiments for training convolutional neural networks for image classification and training a recurrent neural network for time series forecasting are presented. These results show that the algorithm can outperform a stochastic gradient approach and the first-order TRish algorithm in practice.

1 Introduction

For many years, the foundational approach for solving stochastic optimization problems has been the stochastic gradient method ([30]), hereafter referred to as SG. However, despite its many theoretical and practical advantages, there remain some shortcomings in the use of SG for solving many stochastic optimization problems, including many arising in machine learning and signal processing, areas in which SG and its variants are very popular. For example, one disadvantage of SG and many variants of it (see §2) is that the variance of the step taken by the algorithm in each iteration is proportional to the variance of the stochastic gradient estimate, which can be large. In the *fully stochastic* regime, i.e., when the variances of the stochastic gradient estimates are merely bounded by some (large) constant, SG can take a large step even though the norm of the true gradient may be relatively small in norm.

In [13], a first-order stochastic optimization algorithm is proposed that is designed to mitigate the effects of large variances of the stochastic gradient estimates. Based on a trust region methodology, this *trust-region-ish* algorithm, known as TRish, uses a careful step normalization procedure in order to attain theoretical convergence properties on par with those of SG, but in such a way that the empirical performance can be better than that of SG. The results of experiments on logistic regression and deep neural network training problems demonstrates that the empirical performance of TRish can be significantly better than that of a traditional SG approach. In particular, TRish is able to reach better solutions more quickly, and in a more stable manner, meaning that the quality of the solution estimate does not vary wildly from one iteration to the next.

[*] E-mail: frank.e.curtis@lehigh.edu

[†] E-mail: rus415@lehigh.edu

In this paper, we extend the TRish methodology to allow for the use of stochastic second-order information, in the form of stochastic Hessian estimates that are incorporated in the trust region subproblems. The resulting algorithm, which we continue to refer to as TRish, is shown to have good convergence properties in a wide range of settings. In particular, in the fully stochastic regime and with a very loose requirement on the accuracy with which the trust region subproblems are solved, we show that the algorithm achieves convergence properties on par with those of TRish. Admittedly, this is done with assumptions that impose stricter requirements on the stepsizes employed in the algorithm, but the results are still non-trivial to obtain, and the theoretical analysis in this paper requires different techniques than those employed in [13]. We also include some theoretical guarantees that are stronger than have been presented for the first-order variant of TRish. On the other end of the theoretical spectrum, we show that when the stochastic gradient and Hessian estimates are very close in expectation to the true values, and when the subproblems are solved exactly, TRish offers a worst-case complexity property that is similar to that offered by a determinstic second-order trust region method.

As has been the motivation for other authors considering second-order extensions of stochastic optimization algorithms, a main motivation for our work is to design an algorithm that can ideally inherit the benefits of Newton-trust-region methods for minimization, such as their scale invariance, ability to employ problem-independent stepsizes near a solution (as opposed to first-order methods, which require problem-dependent stepsizes, typically related to the Lipschitz constant for the gradient of the objective), ability to handle nonconvexity and avoid saddle points without extra computational procedures, and asymptotic fast rate of convergence. These properties cannot fully be attained in the stochastic regime, but our numerical experiments demonstrate that the TRish methodology can benefit from the use of stochastic second-order derivative information in practice. The results that we present in this paper are for training convolutional neural networks (CNNs) for image classification, and for training a recurrent neural network (RNN) for time series forecasting. Our results suggest that TRish can be an effective approach for stochastic and finite-sum minimization over broad classes of challenging problems.

2 Literature Review

The literature on SG, a stochastic first-order method, is extensive. For a few examples of papers with analyses of SG and variants of it, see [4, 28, 1, 5, 9, 14, 18, 25, 19, 30], and [31].

Stochastic second-order methods, which can be classified as methods that compute each step by (approximately) minimizing a quadratic model of the objective function, have received less attention in the literature. That said, many types of methods have been proposed, analyzed, and tested. Overall, one may characterize stochastic second-order methods into four categories (see [4]): stochastic Newton methods, stochastic quasi-Newton methods, natural gradient methods, and diagonal-scaling methods.

Stochastic Newton methods, like the deterministic Newton method for minimization, compute each step by approximately minimizing a quadratic model of the form $g_k^T s + \frac{1}{2} s^T H_k s$ over $s \in \mathbb{R}^n$, where g_k is a stochastic gradient estimate and H_k is a stochastic Hessian estimate. For practical purposes, such an approach would typically use an iterative method such as the conjugate gradient (CG) algorithm to minimize this quadratic function approximately. In this manner, one need not form nor factor the matrix H_k; instead, one need only perform matrix-vector products with H_k, which can be done with back propagation. (In nonconvex settings, a regularization term might also be added if H_k might not be positive definite, or one might terminate CG once negative curvature is detected, as in the standard Steihaug-CG routine ([34]).) For examples of stochastic Newton methods in the literature, see [2, 6, 4, 3], and [12].

Stochastic quasi-Newton methods borrow the idea from the deterministic optimization literature that, instead of employing second-derivative information, one could derive (inverse) Hessian approximations by observing differences in gradients from one iteration to the next. In the stochastic regime, such an approach needs to have safeguards to account for the fact that the gradients are only estimated in each iteration. For examples of stochastic quasi-Newton methods, see [32, 6, 4, 11], and [36].

Motivated by insights from information geometry, the idea of the natural gradient method is to employ the Fischer information matrix in place of the Hessian when computing a search direction. Due to various simplications that are required to derive a practical algorithm, such an approach reduces to a type of (generalized) Gauss-Newton algorithm. For further information in the natural gradient method and related ideas, see [24, 15, 20], and [38].

Diagonal-scaling methods, wherein each step can be expressed as a diagonal scaling matrix times the negative stochastic gradient, are not always classified as second-order methods. However, we argue, as some others do, that these methods should be viewed in this light, and one can argue that the good performance of such methods in practice is because the algorithms are emulating second-order-type properties. A few popular diagonal scaling methods are RMSprop ([35]), Adagrad ([16]), and Adam ([21] and [29]).

Finally, it should be mentioned that some other types of stochastic trust region methods have appeared in the literature. For a couple of examples, see [8] and [7]. However, these approaches are distinct from the TRish methodology, e.g., due to their need to impose stronger requirements on the stochastic gradient and Hessian estimates to achieve their desired convergence rate guarantees. We also direct the reader to the use of trust regions in reinforcement learning; see, e.g., [33] and [23]. This setting is distinct from the one considered in this paper, but these works provide further evidence of how optimization algorithms based on trust region ideas can be effective in various settings.

3 Algorithm

In this section, we formally present our problem of interest, introduce relevant notation and terminology, and present our proposed algorithm.

3.1 Problem Description

The algorithm that we propose is designed to solve stochastic optimization problems; in particular, it is designed to minimize an objective function $f : \mathbb{R}^n \to \mathbb{R}$ that is defined by an expectation of a function $F : \mathbb{R}^n \times \Xi \to \mathbb{R}$ that depends on a random variable ξ, as in

$$\min_{x \in \mathbb{R}^n} \; f(x), \quad \text{where} \quad f(x) = \mathbb{E}_\xi[F(x, \xi)]. \tag{1}$$

Here, $\mathbb{E}_\xi[\cdot]$ denotes expectation with respect to the distribution of ξ. A related type of problem is one obtained by taking a stochastic average approximation (SAA) of (1). This leads to a finite-sum objective of the form $f(x) = \frac{1}{N}\sum_{i=1}^N f_i(x)$, where $f_i := F(\cdot, \xi_i)$ with ξ_i denoting a realization of the random variable ξ. Our algorithm automatically extends to this setting—whether or not the function arises from an SAA of (1)—where in place of the distribution of ξ one can consider a discrete uniform distribution over $\{1, \dots, N\}$.

The algorithm that we propose makes use of stochastic gradient and stochastic Hessian estimates that, at an algorithm iterate $x_k \in \mathbb{R}^n$, are intended to approximate $\nabla f(x_k)$ and $\nabla^2 f(x_k)$, respectively. These can be understood as follows. First, in the context of (1), a stochastic gradient estimate may be computed as $g_k = \nabla_x F(x_k, \xi_k)$, where ξ_k is a realization of ξ. On the other hand, in the context of minimizing a finite sum, one may consider $g_k = \nabla_x f_{i_k}(x_k)$, where i_k has been generated from a discrete uniform distribution over the index set $\{1, \dots, N\}$. In either setting, g_k could instead represent an average of such quantities and still be thought of as a stochastic gradient estimate. In this case, g_k is commonly referred to as a *mini-batch* estimate. Specifically, for (1) one may consider $g_k = \frac{1}{|\mathcal{S}_k|}\sum_{j \in \mathcal{S}_k} \nabla_x F(x_k, \xi_{k,j})$ and for the finite-sum setting one may consider the estimate $g_k = \frac{1}{|\mathcal{S}_k|}\sum_{j \in \mathcal{S}_k} \nabla_x f_{i_{k,j}}(x_k)$. In the statement of our algorithm, we capture all of these possibilities by writing $g_k \approx \nabla f(x_k)$.

For the stochastic Hessian estimates employed in our algorithm, we write $H_k \approx \nabla^2 f(x_k)$, but in this context the meaning of "estimate" is meant much more loosely. Indeed, in the context of computing g_k, the possibilities in the previous paragraph make sense since our analysis requires that g_k be an unbiased estimator of $\nabla f(x_k)$. However, our assumption on H_k can be much less restrictive. While one might choose in the context of (1) to define $H_k = \nabla^2_{xx} F(x_k, \xi_k^H)$ for some realization ξ_k^H of ξ (or with a mini-batch), most of our analysis merely requires that $\{H_k\}$ is uniformly bounded in norm.

3.2 Algorithm Description

Our algorithm is stated below as TRish. Similar to the first-order version in [13], each iteration involves solving a trust region subproblem involving stochastic derivative estimates. Importantly, for much of our analysis, the algorithm merely requires that each subproblem is solved such that *Cauchy decrease* is achieved. This only requires that the solution vector s_k is feasible and yields a value for the subproblem objective that is at least as good as that offered by the *Cauchy point*, which is the minimizer of the subproblem objective along its steepest descent direction from the origin (subject to the trust region constraint); see [10] or [27]. If $H_k = 0$ for all $k \in \mathbb{N}$, then the algorithm reduces to that in [13]. However, clearly, the algorithm presented here offers much more computational flexibility.

Algorithm TRish : (Second-Order) Trust-Region-ish Algorithm

```
Choose an initial iterate $x_1 \in \mathbb{R}^n$ and positive stepsizes $\{\alpha_k\}$.
Choose positive parameters $\{\gamma_{1,k}\}$ and $\{\gamma_{2,k}\}$ such that $0 < \gamma_{2,k} \leq \gamma_{1,k} < \infty$ for all $k \in \mathbb{N}$.
for all $k \in \mathbb{N}$ do
    Generate a stochastic gradient $g_k \approx \nabla f(x_k)$.
    Compute $s_k$ yielding at least Cauchy decrease for the subproblem
```

$$\min_{s \in \mathbb{R}^n} \; g_k^T s + \frac{1}{2} s^T H_k s \quad \text{s.t.} \quad \|s\|_2 \leq \Delta_k \tag{2}$$

```
    using matrix-vector products with a stochastic Hessian $H_k \approx \nabla^2 f(x_k)$, where
```

$$\Delta_k \leftarrow \begin{cases} \gamma_{1,k}\alpha_k \|g_k\|_2 & \text{if } \|g_k\|_2 \in \left[0, \frac{1}{\gamma_{1,k}}\right) \\ \alpha_k & \text{if } \|g_k\|_2 \in \left[\frac{1}{\gamma_{1,k}}, \frac{1}{\gamma_{2,k}}\right] \\ \gamma_{2,k}\alpha_k \|g_k\|_2 & \text{if } \|g_k\|_2 \in \left(\frac{1}{\gamma_{2,k}}, \infty\right]. \end{cases} \tag{3}$$

```
    Set $x_{k+1} \leftarrow x_k + s_k$.
end for
```

Further motivation for the scheme for choosing the trust region radii, namely, (3), can be found in [13]. In short, if one were merely to choose $\Delta_k = \alpha_k$ for all $k \in \mathbb{N}$ so that the steplength is normalized in all iterations, then one might not have a convergent algorithm; it is possible that the algorithm would compute a direction that is one of expected ascent. An example showing this possibility is shown as [13, Ex. 1]. Hence, (3) embodies a *careful* step normalization strategy that might choose $\Delta_k = \alpha_k$, but otherwise uses a nonlinear stepsize control scheme to adjust the steplength. The specific formulas for the radii in (3) ensure that (in the case $H_k = 0$) the steplength $\|x_{k+1} - x_k\|_2$ is a continuous function of $\|g_k\|_2$; see [13, Fig. 1].

4 Convergence Analysis

We prove convergence results for TRish under various settings. We begin by proving fundamental lemmas under basic sets of assumptions. These results illuminate the critical features of the algorithm that lead to all convergence guarantees. We present these guarantees first for the case of nonconvex f and different stepsize and parameter choices, then for the case of f satisfying the well-known Polyak-Łojasiewicz (PL) condition, of which strongly convex functions are a special case. Again, these results are presented for a few stepsize and parameter choices. As TRish generalizes the first-order algorithm proposed in [13], the convergence theorems proved in this section essentially generalize those results proved for the first-order algorithm. However, the proofs presented here require different approaches due to the influence of $\{H_k\}$ on the subproblems.

For convenience throughout our analysis, we denote for all $k \in \mathbb{N}$ the following cases, which clearly correspond to the different cases for the trust region radius Δ_k in (3):

$$
\begin{aligned}
&\|g_k\| \in \left[0, \tfrac{1}{\gamma_{1,k}}\right), && \text{(Case 1)}\\
&\|g_k\| \in \left[\tfrac{1}{\gamma_{1,k}}, \tfrac{1}{\gamma_{2,k}}\right], && \text{(Case 2)}\\
\text{or}\quad &\|g_k\| \in \left(\tfrac{1}{\gamma_{2,k}}, \infty\right). && \text{(Case 3)}
\end{aligned}
$$

Also, for shorthand, we use $\mathbb{E}_k[\cdot]$ to denote expectation of a random variable conditioned on the event that the algorithm has reached the iterate x_k and generates the stochastic gradient estimate g_k and the stochastic Hessian estimate H_k; i.e.,

$$\mathbb{E}_k[\cdot] \equiv \mathbb{E}[\ \cdot \mid \text{the current iterate and derivative estimates are } x_k,\ g_k, \text{ and } H_k].$$

We make the following assumptions throughout our analysis. These assumptions are essentially the same as the basic assumptions from [13], except that we add the assumption that f is twice continuously differentiable, which is a reasonable assumption to add in the context of a second-order-type algorithm.

Assumption 4.1. *The objective function $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable and bounded below by a scalar $f_{\inf} := \inf_{x\in\mathbb{R}^n} f(x) \in \mathbb{R}$. In addition, the gradient function $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous with constant $L_g \in \mathbb{R}_{>0}$ (i.e., f is L_g-*smooth*).*

It is known (see, e.g., [26, Lemma 1.2.2]), that under Assumption 4.1 one has

$$\|\nabla^2 f(x)\|_2 \le L_g \quad \text{for all} \quad x \in \mathbb{R}^n. \tag{4}$$

Assumption 4.2. *For all $k \in \mathbb{N}$, the stochastic gradient estimate g_k is an unbiased estimator of the gradient $\nabla f(x_k)$ in the sense that $\mathbb{E}_k[g_k] = \nabla f(x_k)$.*

Under Assumption 4.2, one finds for all $k \in \mathbb{N}$ that

$$
\begin{aligned}
\mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2] &= \mathbb{E}_k[\|\nabla f(x_k)\|_2^2 - 2\nabla f(x_k)^T g_k + \|g_k\|_2^2] \\
&= -\|\nabla f(x_k)\|_2^2 + \mathbb{E}_k[\|g_k\|_2^2].
\end{aligned} \tag{5}
$$

4.1 Fundamental Lemmas

Our first lemma provides a bound on the subsequent function value with each step that holds regardless of the properties of the generated stochastic derivative estimates.

Lemma 4.1. *Suppose Assumption 4.1 holds. For all $k \in \mathbb{N}$, for any (g_k, H_k), one finds*

$$f(x_{k+1}) \le f(x_k) + g_k^T s_k + \frac{1}{2} s_k^T H_k s_k + (\nabla f(x_k) - g_k)^T s_k + \frac{1}{2}(L_g + \|H_k\|_2)\|s_k\|_2^2.$$

Proof. Proof. Since f is twice continuously differentiable under Assumption 4.1, it follows by Taylor's theorem that there exists $\hat{x}_k$ on the line segment $[x_k, x_{k+1}]$ such that

$$
\begin{aligned}
f(x_{k+1}) - f(x_k) &= \nabla f(x_k)^T s_k + \frac{1}{2} s_k^T \nabla^2 f(\hat{x}_k) s_k \\
&= g_k^T s_k + \frac{1}{2} s_k^T H_k s_k + (\nabla f(x_k) - g_k)^T s_k + \frac{1}{2} s_k^T (\nabla^2 f(\hat{x}_k) - H_k) s_k.
\end{aligned}
$$

Then, since the Cauchy-Schwarz and triangle inequalities together imply with (4) that

$$s_k^T(\nabla^2 f(\hat{x}_k) - H_k)s_k \le \|\nabla^2 f(\hat{x}_k) - H_k\|_2 \|s_k\|_2^2 \le (L_g + \|H_k\|_2)\|s_k\|_2^2,$$

the desired result follows. □

Our next lemma is a Cauchy decrease result on the reduction in a quadratic model of the objective function yielded by each computed step. This type of result is standard in the literature on trust region methods, so we state it without a detailed proof.

Lemma 4.2. *For all $k \in \mathbb{N}$, for any (g_k, H_k), one finds*

$$g_k^T s_k + \frac{1}{2} s_k^T H_k s_k \leq -\frac{1}{2}\|g_k\|_2 \min\left\{\Delta_k, \frac{\|g_k\|_2}{\|H_k\|_2}\right\}.$$

Proof. Proof. The result follows in the standard manner for Cauchy decrease as in the trust region method literature (see, e.g., [10, Corollary 6.3.2] or [27, Lemma 4.3]) since, whichever subproblem is solved to compute the step s_k, TRish requires that at least Cauchy decrease is attained. □

We also make use of a second Cauchy decrease result, stated below as our third lemma. This lemma is useful only when one adds an additional assumption that the norm of the stochastic Hessian estimate is sufficiently small. We shall add such an assumption for one of our main theorems. (The proof the lemma follows using a similar argument as in the standard proof for Lemma 4.2, but with alternative final steps.)

Lemma 4.3. *For all $k \in \mathbb{N}$, for any (g_k, H_k), one finds*

$$g_k^T s_k + \frac{1}{2} s_k^T H_k s_k \leq -\min\left\{\Delta_k \|g_k\|_2 - \frac{1}{2}\Delta_k^2 \|H_k\|_2, \frac{1}{2}\frac{\|g_k\|_2^2}{\|H_k\|_2}\right\}.$$

Proof. Proof. Using standard analysis for the Cauchy point (see, e.g., [27, Lemma 4.3], one has that the Cauchy point lies in the interior of the trust region constraint if $\|g_k\|_2^3 \leq \Delta_k g_k^T H_k g_k$, and lies on the boundary of the trust region constraint otherwise. If the Cauchy point lies in the interior, then it is given by $s_k^C := -(\|g_k\|_2^2 / g_k^T H_k g_k) g_k$, meaning that, by the Cauchy-Schwarz inequality, the step s_k must satisfy

$$g_k^T s_k + \frac{1}{2} s_k^T H_k s_k \leq g_k^T s_k^C + \frac{1}{2} {s_k^C}^T H_k s_k^C = -\frac{1}{2}\frac{\|g_k\|_2^4}{g_k^T H_k g_k} \leq -\frac{1}{2}\frac{\|g_k\|_2^2}{\|H_k\|_2}.$$

On the other hand, if the Cauchy point lies on the boundary of the trust region constraint, then it is given by $s_k^C := -(\Delta_k / \|g_k\|_2) g_k$ and the step s_k must satisfy

$$g_k^T s_k + \frac{1}{2} s_k^T H_k s_k \leq g_k^T s_k^C + \frac{1}{2} {s_k^C}^T H_k s_k^C = -\Delta_k \|g_k\|_2 + \frac{1}{2}\Delta_k^2 \frac{g_k^T H_k g_k}{\|g_k\|_2^2} \leq -\Delta_k \|g_k\|_2 + \Delta_k^2 \|H_k\|_2.$$

The result follows by combining the conclusions of these two cases. □

Our next lemma shows that if the stepsize parameter α_k is sufficiently small relative to a quantity involving $\gamma_{1,k}$, $\gamma_{2,k}$, and $\|H_k\|_2$, then the expected reduction in the objective function value with each step is bounded by a function of the expected squared norm of the stochastic gradient estimate, the variance of the stochastic gradient estimate, and the algorithm parameters. The bound on the reduction proved here will be refined in various ways later in our analysis as we consider the behavior of the algorithm under different sets of assumptions on the derivative estimates and on the stepsize and parameter sequences.

Lemma 4.4. *Suppose that Assumption 4.1 holds and that, for all $k \in \mathbb{N}$,*

$$0 < \alpha_k \leq \frac{\gamma_{2,k}}{4\gamma_{1,k}^2 (L_g + \|H_k\|_2)}. \tag{6}$$

Then, for all $k \in \mathbb{N}$, one finds

$$\mathbb{E}_k[f(x_{k+1})] \leq f(x_k) - \frac{1}{8}\gamma_{2,k}\alpha_k \mathbb{E}_k[\|g_k\|_2^2] + \frac{\gamma_{1,k}^2}{\gamma_{2,k}}\alpha_k \mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2].$$

Proof. Proof. We divide the proof according to the three cases defined on page 6.

Case 1. By Lemma 4.2, it follows in this case that

$$g_k^T s_k + \frac{1}{2} s_k^T H_k s_k \leq -\frac{1}{2}\|g_k\|_2 \min\left\{\gamma_{1,k}\alpha_k\|g_k\|_2, \frac{\|g_k\|_2}{\|H_k\|_2}\right\}.$$

Since (6) ensures $\gamma_{1,k}\alpha_k \leq \frac{\gamma_{2,k}}{4\gamma_{1,k}(L_g+\|H_k\|_2)} \leq \frac{1}{4(L_g+\|H_k\|_2)} \leq \frac{1}{\|H_k\|_2}$, this implies that

$$g_k^T s_k + \frac{1}{2} s_k^T H_k s_k \leq -\frac{1}{2}\gamma_{1,k}\alpha_k\|g_k\|_2^2.$$

Combining this with the result of Lemma 4.1, the Cauchy-Schwarz inequality, and the fact that $\|s_k\|_2 \leq \gamma_{1,k}\alpha_k\|g_k\|_2$ in this case, one finds that

$$\begin{aligned} & f(x_{k+1}) - f(x_k) \\ & \leq g_k^T s_k + \frac{1}{2}s_k^T H_k s_k + (\nabla f(x_k) - g_k)^T s_k + \frac{1}{2}(L_g + \|H_k\|_2)\|s_k\|_2^2 \\ & \leq\ -\frac{1}{2}\gamma_{1,k}\alpha_k\|g_k\|_2^2 + \|\nabla f(x_k) - g_k\|_2\|s_k\|_2 + \frac{1}{2}(L_g + \|H_k\|_2)\|s_k\|_2^2 \\ & \leq\ -\frac{1}{2}\gamma_{1,k}\alpha_k\|g_k\|_2^2 + \gamma_{1,k}\alpha_k\|\nabla f(x_k) - g_k\|_2\|g_k\|_2 + \frac{1}{2}\gamma_{1,k}^2\alpha_k^2(L_g + \|H_k\|_2)\|g_k\|_2^2. \end{aligned} \tag{7}$$

Since $0 \leq (\frac{1}{2}\|g_k\|_2 - \|\nabla f(x_k) - g_k\|_2)^2 = \frac{1}{4}\|g_k\|_2^2 - \|\nabla f(x_k) - g_k\|_2\|g_k\|_2 + \|\nabla f(x_k) - g_k\|_2^2$ and since (6) implies $\gamma_{1,k}\alpha_k \leq \frac{\gamma_{2,k}}{4\gamma_{1,k}(L_g+\|H_k\|_2)} \leq \frac{1}{4(L_g+\|H_k\|_2)}$, one finds that

$$\begin{aligned} & f(x_{k+1}) - f(x_k) \\ & \leq\ -\frac{1}{2}\gamma_{1,k}\alpha_k\|g_k\|_2^2 + \gamma_{1,k}\alpha_k\left(\frac{1}{4}\|g_k\|_2^2 + \|\nabla f(x_k) - g_k\|_2^2\right) + \frac{1}{2}\gamma_{1,k}^2\alpha_k^2(L_g + \|H_k\|_2)\|g_k\|_2^2 \\ & \leq\ -\frac{1}{8}\gamma_{1,k}\alpha_k\|g_k\|_2^2 + \gamma_{1,k}\alpha_k\|\nabla f(x_k) - g_k\|_2^2, \end{aligned}$$

which implies the desired conclusion since $\gamma_{1,k} \geq \gamma_{2,k}$.

Case 2. By Lemma 4.2 and since in this case one has $\gamma_{2,k}\|g_k\|_2 \leq 1$, it follows that

$$g_k^T s_k + \frac{1}{2}s_k^T H_k s_k \leq -\frac{1}{2}\|g_k\|_2 \min\left\{\alpha_k, \frac{\|g_k\|_2}{\|H_k\|_2}\right\} \leq -\frac{1}{2}\|g_k\|_2 \min\left\{\gamma_{2,k}\alpha_k\|g_k\|, \frac{\|g_k\|_2}{\|H_k\|}\right\}.$$

Since (6) ensures $\gamma_{2,k}\alpha_k \leq \gamma_{1,k}\alpha_k \leq \frac{\gamma_{2,k}}{4\gamma_{1,k}(L_g+\|H_k\|_2)} \leq \frac{1}{4(L_g+\|H_k\|_2)} \leq \frac{1}{\|H_k\|_2}$, this implies that

$$g_k^T s_k + \frac{1}{2}s_k^T H_k s_k \leq -\frac{1}{2}\gamma_{2,k}\alpha_k\|g_k\|_2^2.$$

Combining this with the result of Lemma 4.1, the Cauchy-Schwarz inequality, and the fact that $\|s_k\|_2 \leq \alpha_k$ in this case, one finds that

$$\begin{aligned} & f(x_{k+1}) - f(x_k) \\ & \leq g_k^T s_k + \frac{1}{2}s_k^T H_k s_k + (\nabla f(x_k) - g_k)^T s_k + \frac{1}{2}(L_g + \|H_k\|_2)\|s_k\|_2^2 \\ & \leq\ -\frac{1}{2}\gamma_{2,k}\alpha_k\|g_k\|_2^2 + \|\nabla f(x_k) - g_k\|_2\|s_k\|_2 + \frac{1}{2}(L_g + \|H_k\|_2)\|s_k\|_2^2 \\ & \leq\ -\frac{1}{2}\gamma_{2,k}\alpha_k\|g_k\|_2^2 + \alpha_k\|\nabla f(x_k) - g_k\|_2 + \frac{1}{2}\alpha_k^2(L_g + \|H_k\|_2). \end{aligned}$$

Since $0 \leq \frac{\gamma_{2,k}}{\gamma_{1,k}^2}(\frac{1}{2} - \frac{\gamma_{1,k}^2}{\gamma_{2,k}}\|\nabla f(x_k) - g_k\|_2)^2 = \frac{\gamma_{2,k}}{4\gamma_{1,k}^2} - \|\nabla f(x_k) - g_k\|_2 + \frac{\gamma_{1,k}^2}{\gamma_{2,k}}\|\nabla f(x_k) - g_k\|_2^2$ and since $1 \leq \gamma_{1,k}\|g_k\|_2$ in this case, the above and (6) imply the desired conclusion that

$$\begin{aligned}
&f(x_{k+1}) - f(x_k) \\
\leq\ & -\frac{1}{2}\gamma_{2,k}\alpha_k\|g_k\|_2^2 + \alpha_k\left(\frac{\gamma_{2,k}}{4\gamma_{1,k}^2} + \frac{\gamma_{1,k}^2}{\gamma_{2,k}}\|\nabla f(x_k) - g_k\|_2^2\right) + \frac{1}{2}\alpha_k^2(L_g + \|H_k\|_2) \\
=\ & -\frac{1}{2}\gamma_{2,k}\alpha_k\|g_k\|_2^2 + \frac{1}{4}\gamma_{2,k}\alpha_k\|g_k\|_2^2 + \frac{\gamma_{1,k}^2}{\gamma_{2,k}}\alpha_k\|\nabla f(x_k) - g_k\|_2^2 + \frac{1}{2}\gamma_{1,k}^2\alpha_k^2(L_g + \|H_k\|_2)\|g_k\|_2^2 \\
\leq\ & -\frac{1}{8}\gamma_{2,k}\alpha_k\|g_k\|_2^2 + \frac{\gamma_{1,k}^2}{\gamma_{2,k}}\alpha_k\|\nabla f(x_k) - g_k\|_2^2.
\end{aligned}$$

Case 3. The proof follows in the same manner as the proof for Case 1, using $\gamma_{2,k} \leq \gamma_{1,k}$.

The desired conclusion follows by combining the results for the three cases. □

Our final fundamental lemma proves a similar type of bound on the expected reduction in the objective function as in the preceding lemma, except that it can offer a stronger bound when the difference $\gamma_{1,k} - \gamma_{2,k}$ is proportional to α_k and there is an appropriate balance between the stepsize α_k and the norm of the stochastic Hessian estimate. (Note that to ensure the bound on $\|H_k\|_2$ that is required for the lemma, one might need to scale H_k, causing $\mathbb{E}_k[H_k] \neq \nabla^2 f(x_k)$. This might not seem ideal, but as is known in the deterministic optimization literature, it still allows one to incorporate some (approximate) second-order information, which can be beneficial in practice.) We consider the behavior of the algorithm in such situations in one of our main theorems.

Lemma 4.5. *Suppose that Assumption 4.1 holds and that, for all $k \in \mathbb{N}$ and some $\eta \in \mathbb{R}_{>0}$,*

$$\begin{aligned}
&0 < \alpha_k \leq \min\left\{\frac{\gamma_{2,k}}{4\gamma_{1,k}^2(L_g + \|H_k\|_2)}, \frac{1}{6\eta + 2\gamma_{1,k}(L_g + \|H_k\|_2)}\right\}, \\
&\|H_k\|_2 \leq \frac{\eta}{2\gamma_{1,k}}, \quad \textit{and} \quad \gamma_{1,k} - \gamma_{2,k} = \frac{1}{2}\eta\gamma_{1,k}\alpha_k.
\end{aligned} \tag{8}$$

(For one thing, this ensures that (6) *holds for all $k \in \mathbb{N}$.) Then, for all $k \in \mathbb{N}$, one finds*

$$\begin{aligned}
\mathbb{E}_k[f(x_{k+1})] \leq f(x_k) &- \frac{1}{4}\gamma_{2,k}\alpha_k\|\nabla f(x_k)\|_2^2 \\
&+ \frac{1}{2}\left(3\eta + \gamma_{1,k}(L_g + \|H_k\|_2)\right)\gamma_{1,k}\alpha_k^2\mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2].
\end{aligned}$$

Proof. Proof. We divide the proof according to the three cases defined on page 6.

Case 1. By (6), it follows that $\gamma_{1,k}\alpha_k \leq \frac{\gamma_{2,k}}{4\gamma_{1,k}(L_g+\|H_k\|_2)} \leq \frac{1}{4(L_g+\|H_k\|_2)} \leq \frac{1}{2\|H_k\|_2}$ for all $k \in \mathbb{N}$, meaning that for all $k \in \mathbb{N}$ one finds in this case that

$$\Delta_k\|g_k\|_2 - \frac{1}{2}\Delta_k^2\|H_k\|_2 = \gamma_{1,k}\alpha_k\|g_k\|_2^2 - \frac{1}{2}\gamma_{1,k}^2\alpha_k^2\|g_k\|_2^2\|H_k\|_2 \leq \frac{1}{2}\frac{\|g_k\|_2^2}{\|H_k\|_2},$$

while at the same time $\frac{1}{2}\|H_k\|_2 \leq 2\|H_k\|_2 \leq \frac{\eta}{\gamma_{1,k}}$, meaning for all $k \in \mathbb{N}$ that

$$\begin{aligned}
\Delta_k\|g_k\|_2 - \frac{1}{2}\Delta_k^2\|H_k\|_2 &= \gamma_{1,k}\alpha_k\|g_k\|_2^2 - \frac{1}{2}\gamma_{1,k}^2\alpha_k^2\|g_k\|_2^2\|H_k\|_2 \\
&\geq \gamma_{1,k}\alpha_k\|g_k\|_2^2 - \gamma_{1,k}\alpha_k^2\eta\|g_k\|_2^2 = (1 - \alpha_k\eta)\gamma_{1,k}\alpha_k\|g_k\|_2^2.
\end{aligned}$$

(Observe that (8) ensures that $\alpha < \frac{1}{\eta}$, meaning that $1 - \alpha_k\eta > 0$.) Combining these facts with the results of Lemmas 4.1 and 4.3, the Cauchy-Schwarz inequality, and the fact that $\|s_k\|_2 \leq \gamma_{1,k}\alpha_k\|g_k\|_2$ in this case, one finds that

$$\begin{aligned}
& f(x_{k+1}) - f(x_k) \\
\leq\ & g_k^T s_k + \frac{1}{2} s_k^T H_k s_k + (\nabla f(x_k) - g_k)^T s_k + \frac{1}{2}(L_g + \|H_k\|_2)\|s_k\|_2^2 \\
\leq\ & -(1 - \alpha_k\eta)\gamma_{1,k}\alpha_k\|g_k\|_2^2 + \|\nabla f(x_k) - g_k\|_2\|s_k\|_2 + \frac{1}{2}(L_g + \|H_k\|_2)\|s_k\|_2^2 \\
\leq\ & -(1 - \alpha_k\eta)\gamma_{1,k}\alpha_k\|g_k\|_2^2 + \gamma_{1,k}\alpha_k\|\nabla f(x_k) - g_k\|_2\|g_k\|_2 + \frac{1}{2}\gamma_{1,k}^2\alpha_k^2(L_g + \|H_k\|_2)\|g_k\|_2^2.
\end{aligned}$$

Since $0 \leq \frac{1}{2}(\|g_k\|_2 - \|\nabla f(x_k) - g_k\|_2)^2 = \frac{1}{2}\|g_k\|_2^2 - \|\nabla f(x_k) - g_k\|_2\|g_k\|_2 + \frac{1}{2}\|\nabla f(x_k) - g_k\|_2^2$,

$$\begin{aligned}
& f(x_{k+1}) - f(x_k) \\
\leq\ & -\left(1 - \alpha_k\eta - \frac{1}{2}\gamma_{1,k}\alpha_k(L_g + \|H_k\|_2)\right)\gamma_{1,k}\alpha_k\|g_k\|_2^2 + \frac{1}{2}\gamma_{1,k}\alpha_k(\|g_k\|_2^2 + \|\nabla f(x_k) - g_k\|_2^2) \\
=\ & -\left(\frac{1}{2} - \alpha_k\eta - \frac{1}{2}\gamma_{1,k}\alpha_k(L_g + \|H_k\|_2)\right)\gamma_{1,k}\alpha_k\|g_k\|_2^2 + \frac{1}{2}\gamma_{1,k}\alpha_k\|\nabla f(x_k) - g_k\|_2^2,
\end{aligned}$$

which along with (5) (applied twice) implies that

$$\begin{aligned}
& \mathbb{E}_k[f(x_{k+1})] - f(x_k) \\
\leq\ & -\left(\frac{1}{2} - \alpha_k\eta - \frac{1}{2}\gamma_{1,k}\alpha_k(L_g + \|H_k\|_2)\right)\gamma_{1,k}\alpha_k\mathbb{E}_k[\|g_k\|_2^2] + \frac{1}{2}\gamma_{1,k}\alpha_k(-\|\nabla f(x_k)\|_2^2 + \mathbb{E}_k[\|g_k\|_2^2]) \\
=\ & -\frac{1}{2}\gamma_{1,k}\alpha_k\|\nabla f(x_k)\|_2^2 + \left(\eta + \frac{1}{2}\gamma_{1,k}(L_g + \|H_k\|_2)\right)\gamma_{1,k}\alpha_k^2\mathbb{E}_k[\|g_k\|_2^2] \\
=\ & -\frac{1}{2}\gamma_{1,k}\alpha_k\|\nabla f(x_k)\|_2^2 + \left(\eta + \frac{1}{2}\gamma_{1,k}(L_g + \|H_k\|_2)\right)\gamma_{1,k}\alpha_k^2(\|\nabla f(x_k)\|_2^2 + \mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2]) \\
=\ & -\left(\frac{1}{2} - \left(\eta + \frac{1}{2}\gamma_{1,k}(L_g + \|H_k\|_2)\right)\alpha_k\right)\gamma_{1,k}\alpha_k\|\nabla f(x_k)\|_2^2 \\
& \quad + \left(\eta + \frac{1}{2}\gamma_{1,k}(L_g + \|H_k\|_2)\right)\gamma_{1,k}\alpha_k^2\mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2].
\end{aligned}$$

Hence, with the inequality above, the desired result follows in this case due to the upper bound imposed on α_k and the fact that $\gamma_{1,k} \geq \gamma_{2,k}$ for all $k \in \mathbb{N}$.

Case 2. Under the conditions of the lemma, one has $\alpha_k \leq \frac{1}{\eta}$ and $2\|H_k\|_2 \leq \frac{\eta}{\gamma_{1,k}}$. In addition, in this case, one has $\gamma_{1,k}\|g_k\|_2 \geq 1$. These facts combined imply that

$$\begin{gathered}
\Delta_k\|g_k\|_2 - \frac{1}{2}\Delta_k^2\|H_k\|_2 = \alpha_k\|g_k\|_2 - \frac{1}{2}\alpha_k^2\|H_k\|_2 \leq \frac{\|g_k\|_2}{\eta} \\
\text{while } \ \frac{1}{2}\frac{\|g_k\|_2^2}{\|H_k\|_2} \geq \frac{\gamma_{1,k}\|g_k\|_2^2}{\eta} \geq \frac{\|g_k\|_2}{\eta}.
\end{gathered}$$

By Lemma 4.3 and the facts that $\frac{1}{2}\|H_k\|_2 \leq 2\|H_k\|_2 \leq \frac{\eta}{\gamma_{1,k}}$ and $\gamma_{1,k}\|g_k\|_2 \geq 1$, it follows that

$$\begin{aligned}
g_k^T s_k + \frac{1}{2}s_k^T H_k s_k &\leq -\Delta_k\|g_k\|_2 + \frac{1}{2}\Delta_k^2\|H_k\|_2 \\
&= -\alpha_k\|g_k\|_2 + \frac{1}{2}\alpha_k^2\|H_k\|_2 \\
&\leq -(1 - \alpha_k\eta)\alpha_k\|g_k\|_2.
\end{aligned}$$

Combining this fact with the results of Lemmas 4.1 and 4.3, the Cauchy-Schwarz inequality, and the facts that $\gamma_{2,k}\|g_k\|_2 \leq 1$, $\gamma_{1,k}\|g_k\|_2 \geq 1$, and $\|s_k\|_2 \leq \alpha_k$ in this case, one finds that

$$\begin{aligned}
&f(x_{k+1}) - f(x_k) \\
&\leq g_k^T s_k + \frac{1}{2} s_k^T H_k s_k + (\nabla f(x_k) - g_k)^T s_k + \frac{1}{2}(L_g + \|H_k\|_2)\|s_k\|_2^2 \\
&\leq \, -(1-\alpha_k\eta)\alpha_k\|g_k\|_2 + \|\nabla f(x_k) - g_k\|_2\|s_k\|_2 + \frac{1}{2}(L_g + \|H_k\|_2)\|s_k\|_2^2 \\
&\leq \, -(1-\alpha_k\eta)\gamma_{2,k}\alpha_k\|g_k\|_2^2 + \alpha_k\|\nabla f(x_k) - g_k\|_2 + \frac{1}{2}\alpha_k^2(L_g + \|H_k\|_2) \\
&\leq \, -(1-\alpha_k\eta)\gamma_{2,k}\alpha_k\|g_k\|_2^2 + \gamma_{1,k}\alpha_k\|\nabla f(x_k) - g_k\|_2\|g_k\|_2 + \frac{1}{2}\gamma_{1,k}^2\alpha_k^2(L_g + \|H_k\|_2)\|g_k\|_2^2.
\end{aligned}$$

Since $0 \leq \frac{1}{2}(\|g_k\|_2 - \|\nabla f(x_k) - g_k\|_2)^2 = \frac{1}{2}\|g_k\|_2^2 - \|\nabla f(x_k) - g_k\|_2\|g_k\|_2 + \frac{1}{2}\|\nabla f(x_k) - g_k\|_2^2$,

$$\begin{aligned}
&f(x_{k+1}) - f(x_k) \\
&\leq \, -\left((1-\alpha_k\eta)\gamma_{2,k} - \frac{1}{2}\gamma_{1,k}^2\alpha_k(L_g + \|H_k\|_2)\right)\alpha_k\|g_k\|_2^2 + \frac{1}{2}\gamma_{1,k}\alpha_k\left(\|g_k\|_2^2 + \|\nabla f(x_k) - g_k\|_2^2\right) \\
&= \, -\left((1-\alpha_k\eta)\gamma_{2,k} - \frac{1}{2}\gamma_{1,k} - \frac{1}{2}\gamma_{1,k}^2\alpha_k(L_g + \|H_k\|_2)\right)\alpha_k\|g_k\|_2^2 + \frac{1}{2}\gamma_{1,k}\alpha_k\|\nabla f(x_k) - g\|_2^2,
\end{aligned}$$

which along with (5) (applied twice) implies that

$$\begin{aligned}
&\mathbb{E}_k[f(x_{k+1})] - f(x_k) \\
&\leq \, -\left((1-\alpha_k\eta)\gamma_{2,k} - \frac{1}{2}\gamma_{1,k} - \frac{1}{2}\gamma_{1,k}^2\alpha_k(L_g + \|H_k\|_2)\right)\alpha_k\mathbb{E}_k[\|g_k\|_2^2] \\
&\qquad + \frac{1}{2}\gamma_{1,k}\alpha_k(-\|\nabla f(x_k)\|_2^2 + \mathbb{E}_k[\|g_k\|_2^2]) \\
&= \, -\frac{1}{2}\gamma_{1,k}\alpha_k\|\nabla f(x_k)\|_2^2 + \left(\gamma_{1,k} - \gamma_{2,k} + \left(\eta\gamma_{2,k} + \frac{1}{2}\gamma_{1,k}^2(L_g + \|H_k\|_2)\right)\alpha_k\right)\alpha_k\mathbb{E}_k[\|g_k\|_2^2] \\
&\leq \, -\frac{1}{2}\gamma_{1,k}\alpha_k\|\nabla f(x_k)\|_2^2 \\
&\qquad + \left(\gamma_{1,k} - \gamma_{2,k} + \left(\eta\gamma_{2,k} + \frac{1}{2}\gamma_{1,k}^2(L_g + \|H_k\|_2)\right)\alpha_k\right)\alpha_k(\|\nabla f(x_k)\|_2^2 + \mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2]) \\
&= \, -\left(\frac{1}{2}\gamma_{1,k} - (\gamma_{1,k} - \gamma_{2,k}) - \left(\eta\gamma_{2,k} + \frac{1}{2}\gamma_{1,k}^2(L_g + \|H_k\|_2)\right)\alpha_k\right)\alpha_k\|\nabla f(x_k)\|_2^2 \\
&\qquad + \left(\gamma_{1,k} - \gamma_{2,k} + \left(\eta\gamma_{2,k} + \frac{1}{2}\gamma_{1,k}^2(L_g + \|H_k\|_2)\right)\alpha_k\right)\alpha_k\mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2] \\
&= \, -\left(\frac{1}{2} - \frac{1}{2}\left(3\eta - \eta^2\alpha_k + \gamma_{1,k}(L_g + \|H_k\|_2)\right)\alpha_k\right)\gamma_{1,k}\alpha_k\|\nabla f(x_k)\|_2^2 \\
&\qquad + \frac{1}{2}\left(3\eta - \eta^2\alpha_k + \gamma_{1,k}(L_g + \|H_k\|_2)\right)\gamma_{1,k}\alpha_k^2\mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2].
\end{aligned}$$

Hence, the desired result follows for this case, again due to the upper bound on α_k and the fact that $\gamma_{1,k} \geq \gamma_{2,k}$ for all $k \in \mathbb{N}$.

Case 3. The proof for this case follows in the same manner as the proof for Case 1, where the result for this case has a similar form except with $\gamma_{1,k}$ replaced by $\gamma_{2,k}$. For the proof, it should be noted that $\gamma_{2,k}\alpha_k \leq \gamma_{1,k}\alpha_k \leq \frac{1}{2\|H_k\|_2}$, $\frac{1}{2}\|H_k\|_2 \leq \frac{\eta}{\gamma_{1,k}} \leq \frac{\eta}{\gamma_{2,k}}$, and $\|s_k\|_2 \leq \gamma_{2,k}\alpha_k\|g_k\|_2$.

The desired conclusion follows by combining the results for the three cases. □

Now that these fundamental lemmas have been established, which focus on the behavior of the algorithm over a single iteration, we turn to analyzing the behavior of the algorithm over the entire sequence of iterations. We break our analysis into parts based on different assumptions about the problem function and the stochastic derivative estimates. For simplicity in much of our analysis, we consider the behavior of the algorithm when the parameter sequences $\{\gamma_{1,k}\}$ and $\{\gamma_{2,k}\}$ are constant. In such cases, one could prove similar results that allow the sequences not to be constant, as long as they remain within bounded intervals. We also prove one result showing that, in practice, one might define these sequences to have the same limit point, which in a sense makes the algorithm behave asymptotically like a more standard stochastic Newton-type method.

4.2 General (nonconvex) objective functions

First, we consider the case when the algorithm is employed to minimize an objective function satisfying only Assumptions 4.1 and 4.2, and when the following loose assumption holds about the algorithm parameters, stochastic gradients, and stochastic Hessians.

Assumption 4.3. *The variance of the stochastic gradient estimates and the sequence of stochastic Hessian estimates are both uniformly bounded in the sense that there exist constants $(M_g, M_H) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0}$ such that, for all $k \in \mathbb{N}$,*

$$\mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2] \leq M_g \quad \text{and} \quad \|H_k\|_2 \leq M_H.$$

In addition, $(\alpha_k, \gamma_{1,k}, \gamma_{2,k}) = (\alpha, \gamma_1, \gamma_2)$ for all $k \in \mathbb{N}$, where $\gamma_1 \geq \gamma_2 > 0$ and

$$0 < \alpha \leq \frac{\gamma_2}{4\gamma_1^2(L_g + M_H)},$$

which, in particular, implies that (6) holds for all $k \in \mathbb{N}$.

Combining the result of Lemma 4.4 with Assumption 4.3 leads to the following result showing that the expected average squared norm of the gradient at the iterates is bounded.

Theorem 4.1. *Under Assumptions 4.1, 4.2, and 4.3, `TRish` yields*

$$\mathbb{E}\left[\sum_{k=1}^{K} \|\nabla f(x_k)\|_2^2\right] \leq \left(\frac{8}{\gamma_2 \alpha}\right)(f(x_1) - f_{\text{inf}}) + K\left(\frac{8\gamma_1}{\gamma_2} - 1\right) M_g \tag{9a}$$

$$\text{and} \quad \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} \|\nabla f(x_k)\|_2^2\right] \leq \frac{1}{K}\left(\frac{8}{\gamma_2 \alpha}\right)(f(x_1) - f_{\text{inf}}) + \left(\frac{8\gamma_1^2}{\gamma_2^2} - 1\right) M_g$$
$$\xrightarrow{K\to\infty} \left(\frac{8\gamma_1^2}{\gamma_2^2} - 1\right) M_g. \tag{9b}$$

Proof. Proof. Since Assumption 4.3 ensures that (6) holds for all $k \in \mathbb{N}$, it follows that the result of Lemma 4.4 holds; hence, with parameters as in Assumption 4.3, for all $k \in \mathbb{N}$ one has

$$\mathbb{E}_k[f(x_{k+1})] \leq f(x_k) - \frac{1}{8}\gamma_2 \alpha \mathbb{E}_k[\|g_k\|_2^2] + \frac{\gamma_1^2}{\gamma_2}\alpha \mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2].$$

Hence, due to Assumption 4.3 and (5), it follows for all $k \in \mathbb{N}$ that

$$\begin{aligned}
&\mathbb{E}_k[f(x_{k+1})] - f(x_k) \\
\leq\ & -\frac{1}{8}\gamma_2 \alpha(\|\nabla f(x_k)\|_2^2 + \mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2]) + \frac{\gamma_1^2}{\gamma_2}\alpha \mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2] \\
=\ & -\frac{1}{8}\gamma_2 \alpha \|\nabla f(x_k)\|_2^2 + \alpha\left(\frac{\gamma_1^2}{\gamma_2} - \frac{1}{8}\gamma_2\right)\mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2]
\end{aligned}$$

$$\leq -\frac{1}{8}\gamma_2\alpha\|\nabla f(x_k)\|_2^2 + \alpha\left(\frac{\gamma_1^2}{\gamma_2} - \frac{1}{8}\gamma_2\right)M_g. \tag{10}$$

Taking total expectation, it follows for all $k \in \mathbb{N}$ that

$$\mathbb{E}[f(x_{k+1})] - \mathbb{E}[f(x_k)] \leq -\frac{1}{8}\gamma_2\alpha\mathbb{E}[\|\nabla f(x_k)\|_2^2] + \alpha\left(\frac{\gamma_1^2}{\gamma_2} - \frac{1}{8}\gamma_2\right)M_g,$$

which implies

$$\mathbb{E}[\|\nabla f(x_k)\|_2^2] \leq \left(\frac{8}{\gamma_2\alpha}\right)(\mathbb{E}[f(x_k)] - \mathbb{E}[f(x_{k+1})]) + \left(\frac{8\gamma_1^2}{\gamma_2^2} - 1\right)M_g.$$

Summing this inequality over all $k \in \{1, \dots, K\}$ and using the fact that f is bounded below by $f_{\inf}$ yields (9a), which, in turn, implies (9b). □

Next, we consider the behavior of `TRish` when Assumptions 4.1 and 4.2 hold and when a run of the algorithm satisfies the following assumption involving diminishing stepsizes.

Assumption 4.4. *The variance of each stochastic gradient estimate is proportional to the stepsize and the sequence of stochastic Hessian estimates is uniformly bounded in the sense that there exist constants* $(M_g, M_H) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0}$ *such that, for all* $k \in \mathbb{N}$,

$$\mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2] \leq M_g\alpha_k \quad \text{and} \quad \|H_k\|_2 \leq M_H. \tag{11}$$

In addition, $(\gamma_{1,k}, \gamma_{2,k}) = (\gamma_1, \gamma_2)$ *for all* $k \in \mathbb{N}$ *where* $\gamma_1 \geq \gamma_2 > 0$, *and*

$$\{\alpha_k\} = \left\{\frac{a}{b+k}\right\} \quad \text{for some } (a,b) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0}$$

such that (6) *holds for all* $k \in \mathbb{N}$.

Under this assumption, which is stronger than Assumption 4.3, we obtain the following result, which, not surprisingly, is stronger than the result in Theorem 4.1.

Theorem 4.2. *Under Assumptions 4.1, 4.2, and 4.4,* `TRish` *yields*

$$\lim_{K\to\infty} \mathbb{E}\left[\sum_{k=1}^{K} \alpha_k\|\nabla f(x_k)\|^2\right] < \infty \tag{12a}$$

$$\text{and} \quad \mathbb{E}\left[\frac{1}{\sum_{k=1}^{K}\alpha_k}\sum_{k=1}^{K}\alpha_k\|\nabla f(x_k)\|^2\right] \xrightarrow{K\to\infty} 0. \tag{12b}$$

In addition, it follows that

$$\liminf_{k\to\infty} \|f(x_k)\|_2^2 = 0 \quad \text{with probability 1.} \tag{13}$$

Proof. Proof. Following the same arguments as in the proof of Theorem 4.1, for all $k \in \mathbb{N}$,

$$\mathbb{E}_k[f(x_{k+1})] \leq f(x_k) - \frac{1}{8}\gamma_2\alpha_k\|\nabla f(x_k)\|_2^2 + \left(\frac{\gamma_1^2}{\gamma_2} - \frac{1}{8}\gamma_2\right)M_g\alpha_k^2, \tag{14}$$

which, taking total expectation, implies for all $k \in \mathbb{N}$ that

$$\mathbb{E}[f(x_{k+1})] - \mathbb{E}[f(x_k)] \leq -\frac{1}{8}\gamma_2\alpha_k\mathbb{E}[\|\nabla f(x_k)\|_2^2] + \left(\frac{\gamma_1^2}{\gamma_2} - \frac{1}{8}\gamma_2\right)M_g\alpha_k^2.$$

Rearranging terms and summing over all $k \in \{1, \dots, K\}$, it follows that

$$\frac{1}{8}\gamma_2 \sum_{k=1}^{K} \alpha_k \mathbb{E}[\|\nabla f(x_k)\|_2^2] \leq \sum_{k=1}^{K} (\mathbb{E}[f(x_k)] - \mathbb{E}[f(x_{k+1})]) + \left(\frac{\gamma_1^2}{\gamma_2} - \frac{1}{8}\gamma_2\right) M_g \sum_{k=1}^{K} \alpha_k^2. \tag{15}$$

Since $\sum_{k=1}^{K} (\mathbb{E}[f(x_k)] - \mathbb{E}[f(x_{k+1})]) \leq f(x_1) - f_{\text{inf}} < \infty$ for any $K \in \mathbb{N}$ and since Assumption 4.4 implies $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$, it follows from (15) that (12a) holds. Moreover, dividing (15) by $\sum_{k=1}^{K} \alpha_k$ and since Assumption 4.4 implies $\sum_{k=1}^{\infty} \alpha_k = \infty$, it follows that (12b) holds.

Let us now prove (13). Defining the scalars $\beta_1 := \frac{1}{8}\gamma_2$ and $\beta_2 := \left(\frac{\gamma_1^2}{\gamma_2} - \frac{1}{8}\gamma_2\right) M_g$, it follows from (14) that, for all $k \in \mathbb{N}$, the expected reduction in f satisfies

$$\begin{aligned} & \mathbb{E}_k[f(x_{k+1})] \leq f(x_k) - \beta_1 \alpha_k \|\nabla f(x_k)\|^2 + \beta_2 \alpha_k^2 \\ \Longrightarrow \quad & \mathbb{E}_k[f(x_{k+1})] + \beta_2 \sum_{i=k+1}^{\infty} \alpha_i^2 \leq f(x_k) - \beta_1 \alpha_k \|\nabla f(x_k)\|^2 + \beta_2 \sum_{i=k}^{\infty} \alpha_i^2. \end{aligned}$$

Considering the stochastic processes $\{p_k\}$ and $\{q_k\}$, where, for all $k \in \mathbb{N}$,

$$p_k := \beta_1 \alpha_k \|\nabla f(x_k)\|_2^2 \text{ and } q_k := f(x_k) + \beta_2 \sum_{i=k}^{\infty} \alpha_i^2,$$

it follows from above that, for all $k \in \mathbb{N}$,

$$\mathbb{E}_k[q_{k+1} - f_{\text{inf}}] \leq q_k - f_{\text{inf}} - p_k. \tag{16}$$

One finds from this relationship that $\mathbb{E}[q_k - f_{\text{inf}}] < \infty$ and $\mathbb{E}_k[q_{k+1} - f_{\text{inf}}] \leq q_k - f_{\text{inf}}$ for all $k \in \mathbb{N}$, which with $q_k - f_{\text{inf}} \geq 0$ for all $k \in \mathbb{N}$ implies that $\{q_k - f_{\text{inf}}\}$ is a nonnegative supermartingale. This implies (see, e.g., [17] and similar use in [36]) that there exists q such that $\lim_{k\to\infty} q_k = q$ with probability 1 and $\mathbb{E}[q] \leq \mathbb{E}[q_1]$. From (16), one finds that $\mathbb{E}[p_k] \leq \mathbb{E}[q_k] - \mathbb{E}[q_{k+1}]$, from which it follows that

$$\mathbb{E}\left[\sum_{k=1}^{\infty} p_k\right] < \infty \quad \Longrightarrow \quad \sum_{k=1}^{\infty} \beta_1 \alpha_k \|\nabla f(x_k)\|_2^2 = \sum_{k=1}^{\infty} p_k < \infty \text{ with probability 1.} \tag{17}$$

Since $\sum_{k=1}^{\infty} \alpha_k = \infty$ under Assumption 4.4, the above implies (13). □

To conclude this section, let us prove a result that in part considers the behavior of the algorithm under the following assumption.

Assumption 4.5. *The second moment of the stochastic gradient estimates is uniformly bounded in the sense that there exists a constant $M_{g,2} \in \mathbb{R}_{>0}$ such that, for all $k \in \mathbb{N}$,*

$$\mathbb{E}_k[\|g_k\|_2^2] \leq M_{g,2}.$$

It should be said that Assumption 4.5 is strong since it implies that the variance of the stochastic gradient estimates is smaller at points at which $\|\nabla f(x_k)\|_2$ is large. In particular, under Assumptions 4.2 and Assumption 4.5, it follows (recall (5)) that

$$\mathbb{E}_k[\|g_k\|_2^2] \leq M_g \quad \Longrightarrow \quad \mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2] \leq M_g - \|\nabla f(x_k)\|_2^2.$$

That said, if the iterates of the algorithm happen to remain in a region over which $\|\nabla f(\cdot)\|_2$ is bounded, then it is interesting to note that Assumption 4.5 leads to the following strong result about the behavior of the algorithm. (A result similar to the following was proved for a stochastic quasi-Newton method as [36, Th. 2.6], and our proof borrows from that one. That said, our proof corrects an oversight made in the proof of [36, Th. 2.6] when one considers the negation of a statement of the form (18); in particular, in the negation, one should only assume that the limit does not hold with some positive probability, not with complete certainty.)

Theorem 4.3. *Under Assumptions 4.1, 4.2, 4.4, and 4.5, `TRish` yields*

$$\lim_{k\to\infty} \|\nabla f(x_k)\|_2 = 0 \quad \textit{with probability 1.} \tag{18}$$

Proof. Proof. To derive a contradiction, suppose that (18) does not hold, meaning that with some nonzero probability there exists $\epsilon \in (0,\infty)$ and an infinite index set $\mathcal{K}_1 \subseteq \mathbb{N}$ such that $\|\nabla f(x_k)\|_2 > \epsilon$ for all $k \in \mathcal{K}_1$. On the other hand, from Theorem 4.2 it follows that (13) holds, meaning that with probability one there exists an infinite index set $\mathcal{K}_2$ such that $\|\nabla f(x_k)\|_2 \leq \frac{1}{2}\epsilon$ for all $k \in \mathcal{K}_2$. Together, these facts imply with nonzero probability the existence of index sets $\{m_i\}_{i=1}^{\infty} \subset \mathbb{N}$ and $\{n_i\}_{i=1}^{\infty} \subset \mathbb{N}$ with $m_i < n_i$ for all $i \in \mathbb{N}$ such that

$$\begin{aligned} &\|\nabla f(x_{m_i})\| \geq \epsilon, \quad \|\nabla f(x_{n_i})\| < \tfrac{1}{2}\epsilon, \\ &\qquad\qquad \text{and} \quad \|\nabla f(x_k)\| \geq \tfrac{1}{2}\epsilon \ \text{ for all } \ k \in \{m_i+1,\cdots,n_i-1\}. \end{aligned} \tag{19}$$

For the remainder of the proof, let us condition on the event that (19) holds. With (17),

$$\infty > \sum_{k=1}^{\infty} \alpha_k \|\nabla f(x_k)\|_2^2 \geq \sum_{i=1}^{\infty} \sum_{k=m_i}^{n_i-1} \alpha_k \|\nabla f(x_k)\|_2^2 \geq \epsilon^2 \sum_{i=1}^{\infty} \sum_{k=m_i}^{n_i-1} \alpha_k \ \text{ with probability 1,}$$

meaning that

$$\lim_{i\to\infty} \sum_{k=m_i}^{n_i-1} \alpha_k < \infty \ \text{ with probability 1.} \tag{20}$$

Now notice that, for any $k \in \mathbb{N}$, for any (g_k, H_k), Assumption 4.5 implies

$$\mathbb{E}_k[\|x_{k+1} - x_k\|_2] = \mathbb{E}_k[\|s_k\|] \leq \alpha_k \max\{1, \gamma_1 \mathbb{E}_k[\|g_k\|_2]\} = \alpha_k \max\left\{1, \gamma_1 \sqrt{M_{g,2}}\right\},$$

from which it follows that

$$\mathbb{E}_k[\|x_{n_i} - x_{m_i}\|_2] \leq \max\left\{1, \gamma_1 \sqrt{M_{g,2}}\right\} \sum_{k=m_i}^{n_i-1} \alpha_k.$$

Therefore, with (20), one finds that $\lim_{i\to\infty} \|x_{n_i} - x_{m_i}\|_2 = 0$ with probability 1, which with Lipschitz continuity of ∇f under Assumption 4.1 implies that $\lim_{i\to\infty} \|\nabla f(x_{n_i}) - \nabla f(x_{m_i})\|_2 = 0$ with probability 1. However, this contradictions (19). □

4.3 Objective functions satisfying the Polyak-Łojasiewicz condition

We now consider when the algorithm is employed to minimize an objective function satisfying Assumptions 4.1 and 4.2 along with the Polyak-Łojasiewicz (PL) condition. We state this condition in the form of the following assumption.

Assumption 4.6. *There exists a constant $c \in (0,\infty)$ such that, for all $x \in \mathbb{R}^n$, one has*

$$2c(f(x) - f_{\inf}) \leq \|\nabla f(x)\|_2^2 \quad \textit{for all} \ \ x \in \mathbb{R}^n. \tag{21}$$

Functions satisfying Assumption 4.6 include c-strongly convex functions, but also other nonconvex functions. Assumptions 4.1 and 4.6 combined do not guarantee that f has a minimizer, although they do guarantee that if a stationary point exists then it is a global minimizer with objective value $f_{\inf}$. The PL condition is known as a relatively weak condition under which certain algorithms, such as gradient descent, can enjoy a linear rate of convergence. In this section, we show that the theoretical properties for `TRish` are stronger under the PL condition than they are in the more general situations considered in §4.2.

Our first result shows that if the variance of the stochastic gradient estimates and the stochastic Hessian estimates are both uniformly bounded and the algorithm is run with certain fixed parameter settings, then the expected optimality gap is bounded above by a sequence that converges linearly to a constant proportional to M_g/c. This result is comparable to one that can be proved for SG with a fixed stepsize, for which the limiting constant is also $\mathcal{O}(M_g/c)$; see [4, Theorem 4.6].

Theorem 4.4. *Under Assumptions 4.1, 4.2, 4.3, and 4.6, if $\alpha \leq 4/(\gamma_2 c)$, then with*

$$\theta := 4\left(\frac{\gamma_1^2}{\gamma_2^2} - \frac{1}{8}\right)\frac{M_g}{c} \tag{22}$$

`TRish` *yields*

$$\mathbb{E}[f(x_{K+1})] - f_{\inf} \leq \theta + \left(1 - \frac{1}{4}\gamma_2 c\alpha\right)^K (f(x_1) - f_{\inf} - \theta) \xrightarrow{K\to\infty} \theta.$$

Proof. Proof. As in the proof of Theorem 4.1 (see (10)), it follows for all $k \in \mathbb{N}$ that

$$\mathbb{E}_k[f(x_{k+1})] \leq f(x_k) - \frac{1}{8}\gamma_2\alpha\|\nabla f(x_k)\|_2^2 + \alpha\left(\frac{\gamma_1^2}{\gamma_2} - \frac{1}{8}\gamma_2\right)M_g.$$

Hence, by Assumption 4.6, it follows for all $k \in \mathbb{N}$ that

$$\mathbb{E}_k[f(x_{k+1})] \leq f(x_k) - \frac{1}{4}\gamma_2 c\alpha(f(x_k) - f_{\inf}) + \alpha\left(\frac{\gamma_1^2}{\gamma_2} - \frac{1}{8}\gamma_2\right)M_g.$$

Subtracting $f_{\inf}$ from both sides and taking total expectation, it follows for all $k \in \mathbb{N}$ that

$$\mathbb{E}[f(x_{k+1})] - f_{\inf} \leq \left(1 - \frac{1}{4}\gamma_2 c\alpha\right)(\mathbb{E}[f(x_k)] - f_{\inf}) + \alpha\left(\frac{\gamma_1^2}{\gamma_2} - \frac{1}{8}\gamma_2\right)M_g.$$

Therefore, with θ defined in (22), it follows for all $k \in \mathbb{N}$ that

$$\mathbb{E}[f(x_{k+1})] - f_{\inf} - \theta \leq \left(1 - \frac{1}{4}\gamma_2 c\alpha\right)(\mathbb{E}[f(x_k)] - f_{\inf} - \theta).$$

Applying this bound repeatedly for $k \in \{1, \dots, K\}$ yields the desired result. □

Let us now prove, under similar assumptions as in the previous theorem (in particular with respect to the stochastic gradient and Hessian estimates), that `TRish` can offer sublinear decrease of the expected optimality gap to zero if the stepsizes vanish along with the differences $\{\gamma_{1,k} - \gamma_{2,k}\}$. This is the only theorem that we prove in which we consider a case in which $\{\gamma_{1,k}\}$ and $\{\gamma_{2,k}\}$ are not both constant; in particular, we assume $\{\gamma_{1,k}\}$ is constant, but that $\{\gamma_{2,k}\}$ is not. Other similar results can be proved, say with $\{\gamma_{1,k}\}$ converging to a constant sequence $\{\gamma_{2,k}\}$, or with $\{\gamma_{1,k}\}$ and $\{\gamma_{2,k}\}$ both not constant as long as the sequences remain within a positive interval and the difference sequence is proportional to the stepsize sequence in that $\{\gamma_{1,k} - \gamma_{2,k}\} = \mathcal{O}(\alpha_k)$.

For this theorem only, we consider the following assumption.

Assumption 4.7. *The variance of the stochastic gradient estimates and the sequence of stochastic Hessian estimates are both uniformly bounded in the sense that there exist constants $(M_g, M_H) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0}$ such that, for all $k \in \mathbb{N}$,*

$$\mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2] \leq M_g \quad \textit{and} \quad \|H_k\|_2 \leq M_H.$$

In addition, $\gamma_{1,k} = \gamma_1 > 0$ for all $k \in \mathbb{N}$, and

$$\{\alpha_k\} = \left\{\frac{a}{b+k}\right\} \quad \textit{and} \quad \{\gamma_{2,k}\} = \left\{\gamma_1\left(1 - \frac{1}{2}\eta\alpha_k\right)\right\} \quad \textit{for some } (a, b, \eta) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0}$$

such that (8) *holds for all $k \in \mathbb{N}$.*

Under this assumption, we prove sublinear decrease of the expected optimality gap.

Theorem 4.5. *Under Assumptions 4.1, 4.2, 4.6, and 4.7, if the pair $(a, b) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0}$ is chosen such that $\alpha_k \leq \frac{2}{\gamma_{2,1} c}$ for all $k \in \mathbb{N}$, then for all $k \in \mathbb{N}$ the expected optimality gap satisfies*

$$\mathbb{E}[f(x_k)] - f_{\inf} \leq \frac{\phi}{b + k}, \tag{23}$$

where

$$\phi := \max\left\{(b+1)(f(x_1) - f_{\inf}), \frac{\delta_2 a^2}{\delta_1 a - 1}\right\} \in (0, \infty), \tag{24}$$

with

$$\delta_1 := \frac{1}{2}\gamma_{2,1} c \in \left(0, \frac{1}{\alpha}\right] \quad \textit{and} \quad \delta_2 := \frac{1}{2}(3\eta + \gamma_1(L_g + M_H))\gamma_1 M_g \in (0, \infty). \tag{25}$$

Proof. Proof. By Lemma 4.5, it follows for all $k \in \mathbb{N}$ that

$$\mathbb{E}_k[f(x_{k+1})] \leq f(x_k) - \frac{1}{4}\gamma_{2,1}\alpha_k \|\nabla f(x_k)\|_2^2 + \frac{1}{2}(3\eta + \gamma_1(L_g + M_H))\gamma_1 M_g \alpha_k^2. \tag{26}$$

Hence, by Assumption 4.6, it follows for all $k \in \mathbb{N}$ that

$$\mathbb{E}_k[f(x_{k+1})] \leq f(x_k) - \frac{1}{2}\gamma_{2,1} c \alpha_k (f(x_k) - f_{\inf}) + \frac{1}{2}(3\eta + \gamma_1(L_g + M_H))\gamma_1 M_g \alpha_k^2.$$

Subtracting $f_{\inf}$ from both sides and taking total expectation, it follows for all $k \in \mathbb{N}$ that

$$\mathbb{E}[f(x_{k+1})] - f_{\inf} \leq \left(1 - \frac{1}{2}\gamma_{2,1} c \alpha_k\right)(\mathbb{E}[f(x_k)] - f_{\inf}) + \frac{1}{2}(3\eta + \gamma_1(L_g + M_H))\gamma_1 M_g \alpha_k^2.$$

Let us now prove (23) by induction. First, for $k = 1$, the inequality holds by the definition of ϕ in (24). Now suppose that (23) holds up to $k \in \mathbb{N}$. Then, with (δ_1, δ_2) defined in (25), one finds for iteration $(k+1) \in \mathbb{N}$ that

$$\begin{aligned}
\mathbb{E}[f(x_{k+1})] - f_{\inf} &\leq (1 - \delta_1 \alpha_k)(\mathbb{E}[f(x_k)] - f_{\inf}) + \delta_2 \alpha_k^2 \\
&= \left(1 - \frac{\delta_1 a}{b + k}\right)(\mathbb{E}[f(x_k)] - f_{\inf}) + \frac{\delta_2 a^2}{(b+k)^2} \\
&\leq \left(1 - \frac{\delta_1 a}{b + k}\right)\frac{\phi}{b + k} + \frac{\delta_2 a^2}{(b+k)^2} \\
&= \frac{(b+k)\phi}{(b+k)^2} - \frac{\delta_1 a \phi}{(b+k)^2} + \frac{\delta_2 a^2}{(b+k)^2} \\
&= \frac{(b+k-1)\phi}{(b+k)^2} - \frac{(\delta_1 a - 1)\phi}{(b+k)^2} + \frac{\delta_2 a^2}{(b+k)^2} \\
&\leq \frac{(b+k-1)\phi}{(b+k)^2} \leq \frac{\phi}{b+k+1},
\end{aligned}$$

where the last equation follow from the definition of ϕ in (24) and the last inequality follows from the fact that $(z-1)(z+1) \leq z^2$ for any $z \in \mathbb{R}$. □

TRish can also yield sublinear decrease of the expected optimality gap with fixed parameters. However, this can only be guaranteed with the stronger assumption on the stochastic gradient estimates stipulated in Assumption 4.4 (specifically in (11)).

Theorem 4.6. *Under Assumptions 4.1, 4.2, 4.4, and 4.6, if the pair $(a, b) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0}$ is chosen such that $\alpha_k \leq \frac{4}{\gamma_2 c}$ for all $k \in \mathbb{N}$, then for all $k \in \mathbb{N}$ the expected optimality gap satisfies*

$$\mathbb{E}[f(x_k)] - f_{\inf} \leq \frac{\phi}{b + k}, \tag{27}$$

where

$$\phi := \max\left\{(b+1)(f(x_1) - f_{\inf}), \frac{\delta_2 a^2}{\delta_1 a - 1}\right\} \in (0,\infty),$$

with

$$\delta_1 := \frac{1}{4}\gamma_2 c \in \left(0, \frac{1}{\alpha}\right] \quad \textit{and} \quad \delta_2 = \left(\frac{\gamma_1^2}{\gamma_2^2} - \frac{1}{8}\gamma_2\right) M_g \in (0,\infty).$$

Proof. Proof. As in the proof of Theorem 4.2 (see (14)), it follows for all $k \in \mathbb{N}$ that

$$\mathbb{E}_k[f(x_{k+1})] \leq f(x_k) - \frac{1}{8}\gamma_2\alpha_k\|\nabla f(x_k)\|_2^2 + \left(\frac{\gamma_1^2}{\gamma_2} - \frac{1}{8}\gamma_2\right) M_g\alpha_k^2,$$

Noting that this inequality has the same form as that in (26), the remainder of the proof follows in the same manner as that for Theorem 4.5. □

Finally in this section, let us consider the behavior of the algorithm under the following stronger assumption, which requires that the variance of the stochastic gradient estimates vanishes at a geometric rate. Specifically, consider the following assumption.

Assumption 4.8. *The variances of the stochastic gradient estimates decreases at a geometric rate and the sequence of stochastic Hessian estimates is uniformly bounded in the sense that there exist constants* $(M_g, M_H, \zeta) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times (0,1)$ *such that, for all* $k \in \mathbb{N}$,

$$\mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2^2] \leq M_g\zeta^{k-1} \quad \textit{and} \quad \|H_k\|_2 \leq M_H.$$

In addition, $(\alpha_k, \gamma_{1,k}, \gamma_{2,k}) = (\alpha, \gamma_1, \gamma_2)$ *for all* $k \in \mathbb{N}$, *where* $\gamma_1 \geq \gamma_2 > 0$ *and*

$$0 < \alpha \leq \frac{\gamma_2}{4\gamma_1^2(L_g + M_H)},$$

which, in particular, implies that (6) *holds for all* $k \in \mathbb{N}$.

This assumption leads to the following theorem.

Theorem 4.7. *Under Assumptions 4.1, 4.2, 4.6, and 4.8,* `TRish` *yields*

$$\mathbb{E}[f(x_k)] - f_{\inf} \leq \omega\rho^{k-1}, \tag{28}$$

where

$$\kappa_1 := \frac{1}{8}\gamma_2, \quad \kappa_2 := \left(\frac{\gamma_1^2}{\gamma_2} - \frac{1}{8}\gamma_2\right) M_g, \quad \omega := \max\left\{f(x_1) - f_{\inf}, \frac{\kappa_2}{c\kappa_1}\right\}, \\ \textit{and} \quad \rho := \max\{1 - c\kappa_1\alpha, \zeta\} \in (0,1). \tag{29}$$

Proof. Proof. Using the same arguments as in the beginning of the proof of Theorem 4.1 (specifically leading to (10)), one has for all $k \in \mathbb{N}$ that

$$\mathbb{E}_k[f(x_{k+1})] \leq f(x_k) - \frac{1}{8}\gamma_2\alpha\|\nabla f(x_k)\|_2^2 + \alpha\left(\frac{\gamma_1^2}{\gamma_2} - \frac{1}{8}\gamma_2\right) M_g\zeta^{k-1}.$$

Applying the bound in Assumption 4.6, subtracting $f_{\inf}$ from both sides, and taking total expectation, one finds with (κ_1, κ_2) defined in (29) that, for all $k \in \mathbb{N}$, one has

$$\mathbb{E}[f(x_{k+1})] - f_{\inf} \leq (1 - 2c\kappa_1\alpha)(\mathbb{E}[f(x_k)] - f_{\inf}) + \kappa_2\alpha\zeta^{k-1}.$$

Let us now prove (28) by induction. First, for $k = 1$, the inequality follows by the definition of ω in (29). Then, assuming the inequality holds true for $k \in \mathbb{N}$, one finds from above that

$$\begin{aligned}\mathbb{E}[f(x_{k+1})] - f_{\inf} &\leq (1 - 2c\kappa_1\alpha)\omega\rho^{k-1} + \kappa_2\alpha\zeta^{k-1} \\ &= \omega\rho^{k-1}\left(1 - 2c\kappa_1\alpha + \frac{\kappa_2\alpha}{\omega}\left(\frac{\zeta}{\rho}\right)^{k-1}\right) \\ &\leq \omega\rho^{k-1}\left(1 - 2c\kappa_1\alpha + \frac{\kappa_2\alpha}{\omega}\right) \leq \omega\rho^{k-1}(1 - c\kappa_1\alpha) \leq \omega\rho^k,\end{aligned}$$

which proves that the conclusion holds for $k+1$, as desired. □

5 Complexity Analysis

In this section, we prove a complexity result for `TRish`. While not representing the behavior of the algorithm in the fully stochastic regime, the result does show that if one computes sufficiently accurate gradient and Hessian estimates, then one obtains—with the same algorithm—a worst-case performance that is reminiscent of results that can be proved for certain deterministic algorithms with optimal complexity properties. To keep our result in the stochastic setting, we assume only that the stochastic gradients and Hessians are sufficiently accurate in expectation. Consequently, our theorem is weaker than those that can be proved in the deterministic setting. (If one were to replace the conditional expectations in (30) with computed values, then the same arguments would show that `TRish` yields first-order ϵ-stationarity in at most $\mathcal{O}(\epsilon^{-3/2})$ iterations.)

Assumption 5.1. *The Hessian function $\nabla^2 f : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is Lipschitz continuous with constant $L_H \in \mathbb{R}_{>0}$. In addition, given $\epsilon \in \mathbb{R}_{>0}$, the expected distances of the stochastic gradient and stochastic Hessian estimates from the true gradients and Hessians, respectively, are uniformly bounded with respect to (L_H, ϵ) in the sense that there exist constants $\mu_1 \in (0, \frac{1}{12})$ and $\mu_2 \in (0, \frac{1}{12})$ such that, for all $k \in \mathbb{N}$,*

$$\mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2] \leq \frac{\mu_1}{L_H}\epsilon \quad \textit{and} \quad \mathbb{E}_k[\|\nabla^2 f(x_k) - H_k\|_2] \leq \mu_2\sqrt{\epsilon}. \tag{30}$$

Moreover, for all $k \in \mathbb{N}$, the subproblem (2) is solved to global optimality. Finally, the norms of the stochastic gradients are uniformly bounded above and below in that there exists $(G_{low}, G_{high}) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0}$ such that $G_{low} \leq \|g_k\|_2 \leq G_{high}$ for all $k \in \mathbb{N}$.

Under Assumption 5.1, since the subproblem (2) is solved to global optimality for all $k \in \mathbb{N}$, it follows for all $k \in \mathbb{N}$ that there exists a scalar υ_k such that

$$g_k + (H_k + \upsilon_k I)s_k = 0 \tag{31a}$$

$$H_k + \upsilon_k I \succeq 0 \tag{31b}$$

$$\text{and} \quad 0 \leq \upsilon_k \perp \Delta_k - \|s_k\|_2 \geq 0. \tag{31c}$$

Theorem 5.1. *Suppose Assumptions 4.1, 4.2, and 5.1 hold. In addition, suppose that $(\alpha_k, \gamma_{1,k}, \gamma_{2,k}) = (\alpha, \gamma_1, \gamma_2)$ for all $k \in \mathbb{N}$, where $\gamma_1 \geq \gamma_2 > 0$ and for some constants $(\lambda_1, \lambda_2, \lambda_3) \in (0,1) \times (0,1) \times (0,1)$ satisfying*

$$\lambda_1^2\lambda_2^2 - \frac{\mu_1}{\lambda_3} - \frac{\mu_2}{\lambda_3^2} - \frac{2}{3\lambda_3^3} \geq \frac{1}{6} \tag{32}$$

one has that the parameters employed by `TRish` satisfy

$$\alpha \in \left[\frac{2\lambda_1\sqrt{\epsilon}}{L_H}, \frac{2\sqrt{\epsilon}}{L_H}\right], \quad \gamma_1 \in \left[\frac{\lambda_2}{G_{low}}, \infty\right), \quad \textit{and} \quad \gamma_2 \in \left(0, \frac{1}{\lambda_3 G_{high}}\right]. \tag{33}$$

Then, either $\upsilon_k > \sqrt{\epsilon}$ for all $k \in \{1, \dots, K\}$ where $K = \mathcal{O}(\epsilon^{-3/2})$ and the (conditionally) expected total decrease in f in these iterations is at least the initial optimality gap, i.e.,

$$\sum_{k=1}^{K} \mathbb{E}_k[f(x_k) - f(x_{k+1}) | \upsilon_k > \sqrt{\epsilon}] \geq f_0 - f_{\inf}$$

or for some $\overline{K} = \mathcal{O}(\epsilon^{-3/2})$ one finds that

$$\mathbb{E}_{\overline{K}}[\|\nabla f(x_{\overline{K}+1})\|_2 | \upsilon_{\overline{K}} \leq \sqrt{\epsilon}] \leq \mathcal{O}(\epsilon).$$

Proof. Proof. Under Assumption 5.1, it follows for all $k \in \mathbb{N}$ that

$$f(x_k + s_k) - f(x_k) - \nabla f(x_k)^T s_k - \frac{1}{2} s_k^T \nabla^2 f(x_k) s_k \leq \frac{L_H}{6} \|s_k\|^3 \tag{34a}$$

$$\text{and} \quad \|\nabla f(x_k + s_k) - \nabla f(x_k) - \nabla^2 f(x_k) s_k\| \leq \frac{L_H}{2} \|s_k\|^2. \tag{34b}$$

For the next parts of the proof, we consider two cases. In the first case, we show that if the nonnegative scalar υ_k in the optimality conditions (31) is sufficiently small for some $k \in \mathbb{N}$, then the conditional expectation of the gradient of f at x_{k+1} is at most proportional to ϵ. In the second case, when υ_k is not sufficiently small, we show that the conditional expected decrease in the objective function value is at least proportional to $\epsilon^{3/2}$.

First, suppose that $\upsilon_k \leq \sqrt{\epsilon}$. For all $k \in \mathbb{N}$, it follows from the Cauchy-Schwarz inequality, (34b), (31a), and the trust region constraint in (2) that

$$\begin{aligned} \|\nabla f(x_{k+1})\|_2 &\leq \|\nabla f(x_{k+1}) - \nabla f(x_k) - \nabla^2 f(x_k) s_k\|_2 \\ &\quad + \|\nabla f(x_k) - g_k\|_2 + \|(\nabla^2 f(x_k) - H_k) s_k\|_2 + \|g_k + H_k s_k\|_2 \\ &\leq \frac{L_H}{2} \|s_k\|_2^2 + \|\nabla f(x_k) - g_k\|_2 + \|\nabla^2 f(x_k) - H_k\|_2 \|s_k\|_2 + \upsilon_k \|s_k\|_2 \\ &\leq \frac{L_H}{2} \Delta_k^2 + \|\nabla f(x_k) - g_k\|_2 + \Delta_k \|\nabla^2 f(x_k) - H_k\|_2 + \Delta_k \sqrt{\epsilon}. \end{aligned} \tag{35}$$

Let us now consider the three cases defined on page 6. In Case 1, one has that $\|g_k\|_2 \leq 1/\gamma_1$, meaning $\Delta_k = \gamma_1 \alpha \|g_k\|_2 \leq \alpha$. In Case 2, one has that $\Delta_k = \alpha$. Finally, in Case 3, one has by (33) that $\Delta_k = \gamma_2 \alpha \|g_k\|_2 \leq \gamma_2 \alpha G_{high} \leq \alpha / \lambda_3$. Thus, it follows by (35) that, for all $k \in \mathbb{N}$,

$$\begin{aligned} \|\nabla f(x_{k+1})\|_2 &\leq \frac{L_H}{2} \left(\frac{\alpha}{\lambda_3}\right)^2 + \|\nabla f(x_k) - g_k\|_2 + \frac{\alpha}{\lambda_3} \|\nabla^2 f(x_k) - H_k\|_2 + \frac{\alpha}{\lambda_3} \sqrt{\epsilon} \\ &\leq \left(\frac{1}{\lambda_3^2} + \frac{1}{\lambda_3}\right) \frac{2}{L_H} \epsilon + \|\nabla f(x_k) - g_k\|_2 + \frac{2}{L_H \lambda_3} \|\nabla^2 f(x_k) - H_k\|_2 \sqrt{\epsilon}. \end{aligned}$$

Taking conditional expectation, it follows for all $k \in \mathbb{N}$ that

$$\begin{aligned} &\mathbb{E}_k[\|\nabla f(x_{k+1})\|_2 | \upsilon_k \leq \sqrt{\epsilon}] \\ &\leq \left(\frac{1}{\lambda_3^2} + \frac{1}{\lambda_3}\right) \frac{2}{L_H} \epsilon + \mathbb{E}_k[\|\nabla f(x_k) - g_k\|_2] + \frac{2}{L_H \lambda_3} \mathbb{E}_k[\|\nabla^2 f(x_k) - H_k\|_2] \sqrt{\epsilon} \\ &\leq \left(\left(\frac{1}{\lambda_3^2} + \frac{1}{\lambda_3}\right) \frac{2}{L_H} + \frac{\mu_1}{L_H} + \frac{2\mu_2}{L_H \lambda_3}\right) \epsilon. \end{aligned}$$

Second, suppose that $\upsilon_k > \sqrt{\epsilon}$. For such $k \in \mathbb{N}$, it follows by (31c) that $\|s_k\|_2 = \Delta$. Therefore, by (34a), (31), and the Cauchy-Schwarz inequality, it follows for all $k \in \mathbb{N}$ that

$$f(x_{k+1}) - f(x_k)$$

$$
\begin{aligned}
&\leq \nabla f(x_k)^T s_k + \frac{1}{2} s_k^T \nabla^2 f(x_k) s_k + \frac{L_H}{6}\|s_k\|_2^3 \\
&\leq g_k^T s_k + \frac{1}{2} s_k^T H_k s_k + (\nabla f(x_k) - g_k)^T s_k + \frac{1}{2} s_k^T (\nabla^2 f(x_k) - H_k) s_k + \frac{L_H}{6}\|s_k\|_2^3 \\
&\leq g_k^T s_k + \frac{1}{2} s_k^T H_k s_k + \|\nabla f(x_k) - g_k\|_2 \|s_k\|_2 + \frac{1}{2}\|\nabla^2 f(x_k) - H_k\|_2 \|s_k\|_2^2 + \frac{L_H}{6}\|s_k\|_2^3 \\
&= g_k^T s_k + \frac{1}{2} s_k^T H_k s_k + \Delta_k \|\nabla f(x_k) - g_k\|_2 + \frac{1}{2}\Delta_k^2 \|\nabla^2 f(x_k) - H_k\|_2 + \frac{L_H}{6}\Delta_k^3 \\
&\leq -\frac{1}{2}\upsilon_k \Delta_k^2 + \Delta_k \|\nabla f(x_k) - g_k\|_2 + \frac{1}{2}\Delta_k^2 \|\nabla^2 f(x_k) - H_k\|_2 + \frac{L_H}{6}\Delta_k^3 \\
&\leq -\frac{1}{2}\sqrt{\epsilon}\Delta_k^2 + \Delta_k \|\nabla f(x_k) - g_k\|_2 + \frac{1}{2}\Delta_k^2 \|\nabla^2 f(x_k) - H_k\|_2 + \frac{L_H}{6}\Delta_k^3. \qquad (36)
\end{aligned}
$$

Let us consider the three cases defined on page 6. In Case 1, it follows under Assumption 5.1 and by (33) and the fact that $\|g_k\|_2 \leq 1/\gamma$ that $\Delta_k = \gamma_1 \alpha \|g_k\|_2 \geq \gamma_1 \alpha G_{low} \geq \frac{2\lambda_1\lambda_2}{L_H}\sqrt{\epsilon}$ and $\Delta_k \leq \alpha \leq \frac{2}{L_H}\sqrt{\epsilon}$. In Case 2, one finds that $\Delta_k = \alpha \in \left[\frac{2\lambda_1}{L_H}\sqrt{\epsilon}, \frac{2}{L_H}\sqrt{\epsilon}\right]$. Finally, in Case 3, one finds as before that $\Delta_k \leq \alpha/\lambda_3$, meaning that $\Delta_k \leq \frac{2}{L_H\lambda_3}\sqrt{\epsilon}$. Moreover, one finds by the fact that $\|g_k\|_2 \geq 1/\gamma_2$ in this case that $\Delta_k \geq \alpha \geq \frac{2\lambda_1}{L_H}\sqrt{\epsilon}$. Hence, for all $k \in \mathbb{N}$, one finds

$$\Delta_k \in \left[\frac{2\lambda_1\lambda_2}{L_H}\sqrt{\epsilon}, \frac{2}{L_H\lambda_3}\sqrt{\epsilon}\right].$$

Combining this inclusion with (36) and (32), one finds for all $k \in \mathbb{N}$ that

$$
\begin{aligned}
\mathbb{E}_k[f(x_{k+1})|\upsilon_k > \sqrt{\epsilon}] - f(x_k) &\leq -\frac{2\lambda_1^2\lambda_2^2}{L_H^2}\epsilon^{3/2} + \frac{2\mu_1}{L_H^2\lambda_3}\epsilon^{3/2} + \frac{2\mu_2}{L_H^2\lambda_3^2}\epsilon^{3/2} + \frac{4}{3L_H^2\lambda_3^3}\epsilon^{3/2} \\
&\leq -\frac{2}{L_H^2}\left(\lambda_1^2\lambda_2^2 - \frac{\mu_1}{\lambda_3} - \frac{\mu_2}{\lambda_3^2} - \frac{2}{3\lambda_3^3}\right)\epsilon^{3/2} \\
&\leq -\frac{1}{3L_H^2}\epsilon^{3/2}.
\end{aligned}
$$

Combining the results of these two cases leads to our desired conclusion. In particular, suppose that $\upsilon_k > \sqrt{\epsilon}$ for all $k \in \{1, \ldots, \hat{k}\}$ for some $\hat{k} \in \mathbb{N}$. It follows from above that

$$\sum_{k=1}^{\hat{k}} \mathbb{E}_k[f(x_k) - f(x_{k+1})|\upsilon_k > \sqrt{\epsilon}] \geq \hat{k}\left(\frac{1}{3L_H^2}\right)\epsilon^{3/2}.$$

The left-hand side of this inequality is greater than $f_1 - f_{\inf}$ as long as

$$\hat{k}\left(\frac{1}{3L_H^2}\right)\epsilon^{3/2} \geq f_1 - f_{\inf} \iff \hat{k} \geq 3L_H^2(f_1 - f_{\inf})\epsilon^{-3/2},$$

which shows that the desired inequality is true for some $K = \mathcal{O}(\epsilon^{-3/2})$. □

6 Numerical Experiments

The goal of our numerical experiments is to show that TRish, with stochastic second-order derivative information incorporated, can outperform SG and first-order TRish (i.e., TRish with $H_k = 0$ for all $k \in \mathbb{N}$). In particular, our goal is to show with a few interesting test problems that TRish can offer a better final solution, and offer better stability throughout the optimization process, in the sense that the quality of the solution estimates does not vary as wildly from one iteration to the next as it might for SG.

6.1 Implementation Details

We implemented TRish and SG in Python. All of our test problems involve training neural networks. The problems were implemented using PyTorch, which allows one to use back propagation to compute stochastic gradient estimates and perform matrix-vector products with stochastic Hessian estimates. For TRish, we implemented a Steihaug-CG routine (see [34]) for approximately solving the trust region subproblems, where for each subproblem the same batch of data samples used to define the stochastic gradient estimate is used to define the stochastic Hessian estimate. To ensure that TRish did not expend too much effort solving any single subproblem, we imposed a limit of 3 on the number of CG iterations performed when solving each subproblem. In our comparisons, we equate the cost of one stochastic gradient estimate with the cost of computing one stochastic-Hessian-vector product. This allows SG and first-order TRish to perform more optimization iterations per epoch than TRish is able to perform.

6.2 Hyperparameters Tuning

The hyperparameters for all algorithms were tuned using a similar approach to that used in [13]. In particular, for each test problem, we proceeded as follows. First, to establish a baseline for the hyperparameter values, we ran SG with a fixed stepsize of $\alpha = 0.1$ and computed G as the average norm of the stochastic gradient estimates computed throughout the run. We then established sets of possible hyperparameter values with the formulas $\alpha = 10^{\lambda}$, $\gamma_1 = \frac{2^a}{G}$, and $\gamma_2 = \frac{1}{2^b G}$, where λ, a, and b were evenly distributed in some interval. (Different intervals were used for each test problem so that, e.g., the best stepsize for SG was never at the extreme of the allowed range. Details are given in the following subsections for each test problem.) For simplicity, we only consider the behavior of the algorithms with fixed hyperparameter values. To ensure that all algorithms were tuned with the same amount of effort, we fixed the total number of hyperparameter settings to be the same for all algorithms. For example, if (first-order) TRish considers 4 values of α, 3 values of γ_1, and 3 values of γ_2, then we allowed SG to consider $4 \times 3 \times 3 = 36$ stepsizes.

To choose the best hyperparameter values for each algorithm for each test problem, we used a standard type of cross validation procedure. Each dataset came equipped with a training set and a testing set of data. We began by randomly selecting points from the training set to form a validation set. For each hyperparameter setting, we ran each algorithm and observed its performance in terms of final validation accuracy (in the case of image classification) or final validation loss (in terms of time series forecasting). Once the best hyperparameter setting was found in this manner, we ran the algorithm using this setting on *all* of the original training data. In the subsections below, we provide plots of the accuracy and/or loss during this final run for the training and testing data.

6.3 `FashionMNIST`

The first dataset that we considered was `FashionMNIST` ([37]). This consists of images of 10 different types of clothing. Each image is a color image of size 28 × 28. There are 60000 training images and 10000 testing images. We randomly chose 10000 images out of the training set as our validation set, and chose the best set of hyperparameters for each algorithm as the one yielding highest classification accuracy on the validation set.

The neural network that we considered for performing classification for this dataset was composed of two convolutional layers (involving 10 and 20 output channels, respectively, with kernal size 5) followed by a dropout layer and three fully connected layers. ReLU activation was used at each hidden layer and the objective is defined using the logistic loss (cross entropy) function. It is known that one can achieve better classification accuracy on `FashionMNIST` using a more sophisticated neural network, but this network offers sufficiently good results in order for us to demonstrate the behavior of TRish.

We ran each algorithm for 5 epochs with a mini-batch size of 128. During tuning, we obtained $G = 1.5644$. For TRish and first-order TRish, we considered 8 stepsize values over $[0.1, 1]$, namely, $\alpha = 10^{-1+i/7}$ for $i \in \{0, 1, \ldots, 7\}$, along with $\gamma_1 \in \{\frac{4}{G}, \frac{16}{G}\} = \{2.5568, 10.2274\}$ and $\gamma_2 \in \{\frac{1}{2G}, \frac{1}{8G}\} = \{0.3196, 0.07990\}$. For a fair comparison (see [13]), this means that it was appropriate to allow SG to consider 32 stepsize choices in the range $[\frac{1}{8G} \times 10^{-1}, \frac{16}{G} \times 10^{0}] = [10^{-2.0974}, 10^{1.0097}] = [0.00799, 10.2275]$. TRish ended up with the values $(\alpha, \gamma_1, \gamma_2) = (0.1930, 10.2274, 0.07990) = (10^{-5/7}, \frac{16}{1.5644}, \frac{1}{8(1.5644)})$, first-order TRish ended up with the values $(\alpha, \gamma_1, \gamma_2) = (0.3727, 2.5568, 0.3196) = (10^{-3/7}, \frac{4}{1.5644}, \frac{1}{2(1.5644)})$, and SG ended up with the value $\alpha = 0.4192 = 10^{-0.3775}$.

Once the hyperparameter values were determined, we ran the algorithms on the training data 5 times each. In Figure 1, we plot the training loss and testing accuracy over the 5 epochs. The line for each algorithm for each plot shows the mean values over the 5 runs with the shaded region showing one standard deviation above and below the mean. One finds that while the first-order algorithms have an edge in the early parts of the runs, eventually TRish overtakes both of the other algorithms in terms of final training loss (for which lower is better) and final testing accuracy (for which higher is better).

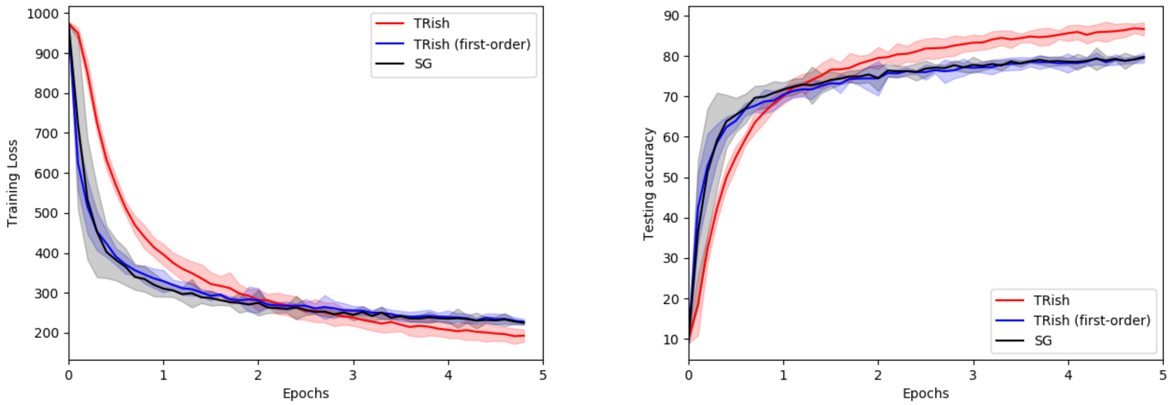

Figure 1: Training loss and testing accuracy during the first five epochs when `TRish`, first-order TRish, and SG are employed to train a convolutional neural network over the `FashionMNIST` dataset.

6.4 `CIFAR-10`.

The second dataset that we considered was `CIFAR-10` ([22]). This dataset consists of 10 classes of color images of different objects. Each image has size 32 × 32. There are 50000 training images and 10000 testing images. We randomly chose 5000 of the training images to compose the validation set. As in the previously subsection, the best set of hyperparameters for each algorithm was chosen as the one yielding highest classification accuracy on the validation set.

The neural network that we considered for this dataset was composed of two convolutional layers (involving 6 and 16 output channels, respectively, with kernal size 5) followed by a max pooling layer, a dropout layer, and three fully connected layers. ReLU activation was used at each hidden layer and the objective was again the logistic loss function. Again, one can achieve better testing accuracy using a more sophisticated neural network, but this network gave sufficiently good results to demonstrate the behavior of our algorithm.

We ran 5 epochs with a mini-batch size of 128. We obtained $G = 2.7819$ and considered $\alpha = 10^{-1+i/7}$ for $i \in \{0, 1, \ldots, 7\}$, $\gamma_1 \in \{\frac{4}{G}, \frac{16}{G}\} = \{1.4378, 5.7515\}$, and $\gamma_2 \in \{\frac{1}{4G}, \frac{1}{80G}\} = \{0.08986, 0.004493\}$. This means that SG was tuned with 32 choices of α in the range $[\frac{1}{80G} \times 10^{-1}, \frac{16}{G} \times 10^{0}] = [10^{-3.3474}, 10^{0.7598}] = [0.0004493, 5.7515]$. TRish chose $(\alpha, \gamma_1, \gamma_2) = (0.1389, 5.7515, 0.004493) = (10^{-6/7}, \frac{16}{2.7819}, \frac{1}{80(2.7819)})$, first-order TRish chose $(\alpha, \gamma_1, \gamma_2) = (0.3727, 5.7515, 0.08986) = (10^{-3/7}, \frac{16}{2.7819}, \frac{1}{4(2.7819)})$, and SG chose $\alpha = 0.2316 = 10^{-0.6352}$.

Figure 2 shows the result of this experiment over 5 runs. Interestingly, for this problem, TRish does not outperform the others in terms of training loss; indeed, first-order TRish appears to give the best results in terms of training loss. However, TRish eventually offers better testing accuracy. While one cannot guarantee that such would be the behavior in general, one does see benefits of TRish-based methods compared to SG.

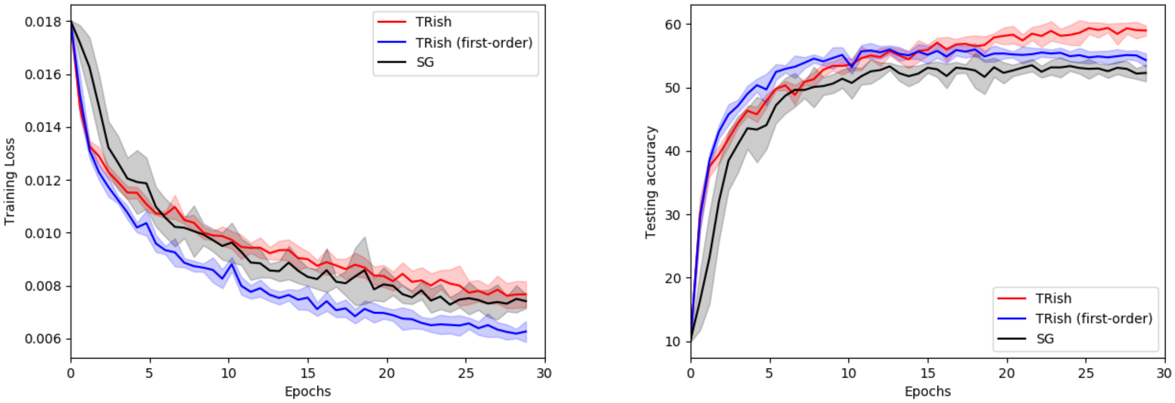

Figure 2: Training loss and testing accuracy during the first five epochs when `TRish`, first-order TRish and SG are employed to train a convolutional neural network over the `CIFAR10` dataset.

6.5 NSW2016

As a final test problem, we considered one of time series forecasting. For this, we used historical data posted online by the Australian Energy Market Operator (AEMO) on demand for electricity in New South Wales in 2016.[1] This gives a univariate time series of length 17423. We used the first 17000 values for our experiments. We used the first 12000 as the training set, the following 2000 as the validation set, and the remaining 3000 as the testing set. We chose the set of hyperparameters that yielded the lowest validation loss.

The recurrent neural network that we considered for this dataset was composed of a single long short-term memory (LSTM) layer with hidden size 32 followed by a fully connected layer. A time step of 10 was used with ReLU activation after the LSTM layer. The objective function used was the mean squared error.

We ran the experiment for 20 epochs using a mini-batch size of 100. We obtained $G = 720.1389$ and considered $\alpha = 10^{-1+i/3}$ for $i = \{0, 1, \dots, 6\}$ along with $\gamma_1 \in \{\frac{4}{G}, \frac{16}{G}\} = \{0.005555, 0.02222\}$ and $\gamma_2 \in \{\frac{1}{2G}, \frac{1}{20G}\} = \{0.0006944, 0.00006944\}$. SG was tuned with 16 choices of α in the range $[\frac{1}{20G} \times 10^{-1}, \frac{16}{G} \times 10^{1}] = [0.000006944, 0.2222] = [10^{-5.1586}, 10^{-0.6532}]$. As a result of hyperparameter tuning, TRish chose $(\alpha, \gamma_1, \gamma_2) = (2.1544, 0.2222, 0.00006944) = (10^{1/3}, \frac{16}{720.1389}, \frac{1}{20(720.1389)})$, first-order TRish chose $(\alpha, \gamma_1, \gamma_2) = (0.4641, 0.005555, 0.0006944) = (10^{-1/3}, \frac{4}{720.1389}, \frac{1}{2(720.1389)})$, and SG chose $\alpha = 0.0002204 = 10^{-3.6567}$.

Figure 3 shows the result of this experiment over 50 runs. The losses are plotted on a logarithmic scale for better viewing of the differences. From the plots, it is clear that while all algorithms reach solutions of comparable quality, TRish is able to achieve low losses earlier than the other two methods, and first-order TRish similarly outperforms SG.

7 Conclusion

A stochastic second-order trust region algorithm has been proposed, analyzed, and tested. It can be viewed as a second-order extension of the algorithm proposed in [13]. We proved theoretical guarantees for the method that are on par with those proved for the first-order algorithm in [13], and in turn comparable to those possessed by SG and many of its variants. That said, our numerical experiments demonstrate that

[1] https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Data-dashboard

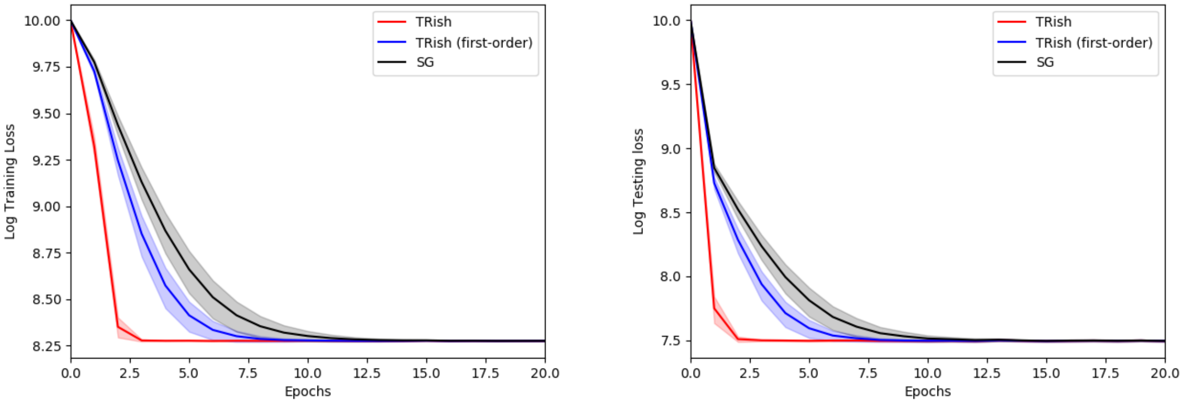

Figure 3: Training loss and testing loss during the first twenty epochs when `TRish`, first-order TRish, and SG are employed to train a recurrent neural network over the `NSW2016` dataset.

the algorithm can perform better in practice, in terms of reaching better solutions and with more stable behavior. We attribute this better behavior to the algorithm's use of carefully chosen trust region radii and stochastic second-order information.

References

[1] A. Agarwal and L. Bottou. A lower bound for the optimization of finite sums. In *Proceedings of the International Conference on Machine Learning*, volume 37, pages 78–86. PMLR, 2015.

[2] Dan Anbar. A stochastic Newton-Raphson method. *Journal of Statistical Planning and Inference*, 2(2):153–163, 1978.

[3] 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, 2018.

[4] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization Methods for Large-Scale Machine Learning. *SIAM Review*, 60(2):223–311, 2018.

[5] R. H. Byrd, G. M. Chin, J. Nocedal, and Y. Wu. Sample Size Selection in Optimization Methods for Machine Learning. *Mathematical Programming, Series B*, 134(1):127–155, 2012.

[6] 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.

[7] Coralia Cartis and Katya Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Mathematical Programming*, 169:1–39, 2018.

[8] Ruobing Chen, Matt Menickelly, and Katya Scheinberg. Stochastic optimization using a trust-region method and random models. *Mathematical Programming*, 169(2):447–487, 2018.

[9] K. L. Chung. On a stochastic approximation method. *Annals of Mathematical Statistics*, 25(3):463–483, 1954.

[10] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 2000.

[11] F. E. Curtis. A Self-Correcting Variable-Metric Algorithm for Stochastic Optimization. In *Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, USA, 2016. JMLR.

[12] F. E. Curtis and D. P. Robinson. Exploiting Negative Curvature in Deterministic and Stochastic Optimization. *Mathematical Programming, Series B*, 176(1):69–94, 2019.

[13] F. E. Curtis, K. Scheinberg, and R. Shi. A Stochastic Trust Region Algorithm Based on Careful Step Normalization. *INFORMS Journal on Optimization*, https://doi.org/10.1287/ijoo.2018.0010, 2019.

[14] C. Dang and G. Lan. Stochastic block mirror descent methods for nonsmooth and stochastic optimization. *SIAM Journal on Optimization*, 25(2):856–881, 2015.

[15] Guillaume Desjardins, Karen Simonyan, Razvan Pascanu, et al. Natural neural networks. In *Advances in Neural Information Processing Systems*, pages 2071–2079, 2015.

[16] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[17] R. Durrett. *Probability: Theory and Examples*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Fifth edition, 2019.

[18] M. P. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data-fitting. *SIAM Journal on Scientific Computing*, 34:A1380–A1405, 2012.

[19] Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[20] Roger Grosse and James Martens. A Kronecker-Factored approximate fisher matrix for convolution layers. In *International Conference on Machine Learning*, pages 573–582, 2016.

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

[22] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical Report, University of Toronto, 2009.

[23] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. *arXiv preprint arXiv:1802.10592*, 2018.

[24] James Martens. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

[25] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust Stochastic Approximation Approach to Stochastic Programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[26] Yu. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer Science+Business Media, LLC, First edition, 2004.

[27] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, second edition, 2006.

[28] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1571–1578. Omnipress, 2012.

[29] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[30] H. Robbins and S. Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[31] H. Robbins and D. Siegmund. A convergence theorem for nonnegative almost supermartingales and some applications. In Jagdish S. Rustagi, editor, *Optimizing Methods in Statistics*. Academic Press, 1971.

[32] 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.

[33] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

[34] Trond Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

[35] 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.

[36] X. Wang, S. Ma, D. Goldfarb, and W. Liu. Stochastic Quasi-Newton Methods for Nonconvex Stochastic Optimization. *SIAM Journal on Optimization*, 27(2):927–956, 2017.

[37] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[38] Guodong Zhang, James Martens, and Roger Grosse. Fast convergence of natural gradient descent for overparameterized neural networks. *arXiv preprint arXiv:1905.10961*, 2019.