SIAM J. OPTIM.
Vol. 29, No. 1, pp. 343–368

A NEWTON-BASED METHOD FOR NONCONVEX OPTIMIZATION WITH FAST EVASION OF SADDLE POINTS[*]

SANTIAGO PATERNAIN[†], ARYAN MOKHTARI[‡], AND ALEJANDRO RIBEIRO[†]

Abstract. Machine learning problems such as neural network training, tensor decomposition, and matrix factorization require local minimization of a nonconvex function. This local minimization is challenged by the presence of saddle points, of which there can be many and from which descent methods may take an inordinately large number of iterations to escape. This paper presents a second-order method that modifies the update of Newton's method by replacing the negative eigenvalues of the Hessian by their absolute values and uses a truncated version of the resulting matrix to account for the objective's curvature. The method is shown to escape saddles in at most $1 + \log_{3/2}(\delta/2\varepsilon)$ iterations, where ε is the target optimality and δ characterizes a point sufficiently far away from the saddle. This base of this exponential escape is 3/2 independently of problem constants. Adding classical properties of Newton's method, the paper proves convergence to a local minimum with probability $1 - p$ in $O(\log(1/p)) + O(\log(1/\varepsilon))$ iterations.

Key words. smooth nonconvex unconstrained optimization, line-search methods, second-order methods, Newton-type methods

AMS subject classifications. 49M05, 49M15, 49M37, 90C06, 90C30

DOI. 10.1137/17M1150116

1. Introduction.

Although it is generally accepted that the distinction between functions that are easy and difficult to minimize is their convexity, a more accurate statement is that the distinction lies in the ability to use local descent methods. A convex function is easy to minimize because a minimum can be found by following local descent directions, but this is not possible for nonconvex functions. This is unfortunate because many interesting problems in machine learning can be reduced to the minimization of nonconvex functions [20]. Despite this general complexity, some recent results have shown that for a large class of nonconvex problems such as dictionary learning [36], tensor decomposition [14], matrix completion [15], and training of some specific forms of neural networks [22], all local minimizers are global minima. This reduces the problem of finding *the* global optimum to the problem of finding *a* local minimum which can be accomplished with local descent methods.

Conceptually, finding a local minimum of a nonconvex function is not more difficult than finding the minimum of a convex function. It is true that the former can have saddle points that are attractors of gradient fields for some initial conditions [27, section 1.2.3]. However, since these initial conditions lie in a low dimensional manifold, gradient descent can be shown to converge almost surely to a local minimum if the initial condition is assumed randomly chosen [24, 30], or if noise is added to gradient descent steps [32]. These fundamental facts notwithstanding, practical implementations show that finding a local minimum of a nonconvex function is much more challenging than finding the minimum of a convex function. This happens be-

[*]Received by the editors October 2, 2017; accepted for publication (in revised form) November 6, 2018; published electronically January 24, 2019.
http://www.siam.org/journals/siopt/29-1/M115011.html
Funding: This work was supported by the ARL DCIST CRA W911NF-17-2-0181.

[†]Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 (spater@seas.upenn.edu, aribeiro@seas.upenn.edu).
[‡]Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 (aryanm@mit.edu).

cause the performance of first order methods is degraded by ill-conditioning which in the case of nonconvex functions implies that it may take a very large number of iterations to escape from a saddle point [12, 7]. Indeed, it can be argued that it is saddle points and not local minima that provide a fundamental impediment to rapid high dimensional nonconvex optimization [12, 2, 35, 34].

In this paper we propose the nonconvex Newton (NCN) method to accelerate the speed of escaping saddles. NCN uses a descent direction analogous to the Newton step except that we use the *positive definite truncated (PT)-inverse* of the Hessian in lieu of the regular inverse of the Hessian (Definition 2.1). The PT-inverse has the same eigenvector basis of the regular inverse, but its eigenvalues differ in the following properties: (i) All negative eigenvalues are replaced by their absolute values. (ii) Small eigenvalues are replaced by a constant. The idea of using the absolute value of the eigenvalues of the Hessian in nonconvex optimization was first proposed in [17] and revisited in [16], [29, Chapters 4 and 7], and [26, 31]. These properties ensure that the value of the function is reduced at each iteration with an appropriate selection of the step size. Our main contribution is to show that NCN can escape any saddle point with eigenvalues bounded away from zero at an exponential rate which can be further shown to have a base of $3/2$ independent of the function's properties in a neighborhood of the saddle. Specifically, we show the following result: (i) Consider an arbitrary $\varepsilon > 0$ and the region around a saddle at which the objective gradient is smaller than ε. There exists a subset of this region so that NCN iterations result in the norm of the gradient growing from ε to δ at an exponential rate with base $3/2$. The number of NCN iterations required for the gradient to progress from ε to δ is therefore not larger than $1 + \log_{3/2}(\delta/2\varepsilon)$; see Theorem 2.2.

We emphasize that the base of escape $3/2$ is independent of the function's properties aside from the requirement to have nondegenerate saddles. The constant δ depends on Lipschitz constants and Hessian eigenvalue bounds. As stated in (i), the base $3/2$ for exponential escape does not hold for all points close to the saddle but for a specific subset at which the gradient norm is smaller than ε. It is impossible to show that NCN iterates stay within this region as they approach the saddle, but we show that it is possible to add noise to NCN iterates to quickly enter into this subset with overwhelming probability. Specifically, we also show the following result: (ii) By adding Gaussian noise with standard deviation proportional to ε when the norm of the gradient of the function is smaller than ε, the region in which the base of the exponential escape of NCN is $3/2$ is visited by the iterates with probability $1 - p$ in $O(\log(1/p))$ iterations. Once this region is visited once, result (i) holds, and we escape the saddle in not more than $1 + \log_{3/2}(\delta/2\varepsilon)$ iterations; see Proposition 3.6.

Combined with other standard properties of the classical Newton's method, results (i) and (ii) imply convergence to a local minimum with probability $1 - p$ in a number of iterations that is of order $O(\log(1/\varepsilon))$ with respect to the target accuracy and of order $O\left(\log(1/p)\right)$ with respect to the desired probability (Theorem 2.3). These convergence rate results are analogous to the results for gradient descent with noise [14, 21]. The fundamental difference is that while gradient descent escapes saddles at an exponential rate with a base that depends on the problem's condition number, NCN escapes saddles at an exponential rate with a base of $3/2$ for all nondegenerate saddles (section 2.2). Section 4 considers the problem of matrix factorization to support theoretical conclusions.

1.1. Related work. Gradient descent for nonconvex functions converges to an ε neighborhood of a critical point, which could be a saddle or a local minimum, in

$O(1/\varepsilon^2)$ iterations [27]. Escaping saddle points is therefore a fundamental problem for which several alternatives have been developed. A line of work in this regard consists in adding noise when entering a neighborhood of the stationary point. The addition of noise ensures that with high probability the iterates will be at a distance sufficiently large from the stable manifold of the saddle, hence reaching the fundamental conclusion that escape from the saddle point can be achieved in $O(\log(1/\varepsilon))$ [14, 21] iterations. Noisy gradient descent therefore converges to an epsilon local minimum in $O(1/\varepsilon^2)$ iterations, matching the rate of convergence of gradient descent to stationary points. Under assumptions of nondegeneracy the iterations needed to converge to a local minimum is $O(\log(1/\varepsilon))$. Although the rate of convergence and the rate of escape from saddles match the corresponding rates for NCN, NCN escapes saddles with an exponential base 3/2, but gradient descent escapes saddles with an exponential rate dependent on the condition number. This difference is very significant in practice (sections 2.2 and 4).

A second approach to ensure that the stationary point attained by the local descent method is a local minimum utilizes second order information to guarantee that the stationary point is a local minimum. These include cubic regularization [18, 28, 5, 6, 1] and trust region algorithms [25, 33, 9, 12, 11], as well as approaches where the descent is computed only along the direction corresponding to the negative eigenvalues of the Hessian [10]. When using a cubic regularization of a second order approximation of the objective the number of iterations needed to converge to an epsilon local minimum—defined as the point for which the norm of the gradient is smaller than ϵ and whose leftmost eigenvalue is $\sqrt{\varepsilon}$—can be further shown to be of order $O(1/\varepsilon^{1.5})$ [28]. Solving this cubic regularization is in itself computationally prohibitive. This is addressed with trust region methods that reduce the computational complexity and still converge to a local minimum in $O(1/\varepsilon^{1.5})$ iterations [11]. A related attempt utilizes low-complexity Hessian-based accelerations to achieve convergence in $O(1/\varepsilon^{7/4})$ iterations [1, 4]. Although these convergence rates seem to be worse than the rate $O(\log(1/\varepsilon))$ achieved by NCN, this is simply a difference in assumptions because we assume here that saddles are nondegenerate. This assumption is absent from [18, 28, 5, 6, 1, 9, 12, 11, 10], and in that sense the theoretical guarantees attained by the algorithm proposed do not hold in some problems of interest. If there were degenerate saddles, our algorithm would converge to one of them in $O(1/\varepsilon^2)$ iterations as well.

2. Nonconvex Newton (NCN) method.

Given a multivariate nonconvex function $f : \mathbb{R}^n \to \mathbb{R}$, we would like to solve the following problem:

$$\mathbf{x}^* := \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x}). \tag{1}$$

Finding $\mathbf{x}^*$ is NP hard in general, except in some particular cases, e.g., when all local minima are known to be global. We then settle for the simpler problem of finding a strong local minima $\mathbf{x}^\dagger$, which we define—when f is twice differentiable and Morse—as any point where the gradient is null and the Hessian is positive definite:

$$\left\|\nabla f(\mathbf{x}^\dagger)\right\| = 0, \qquad \nabla^2 f(\mathbf{x}^\dagger) \succ 0. \tag{2}$$

The fundamental difference between (strongly) convex and nonconvex optimization is that any local minimum is global because there is only one point at which $\|\nabla f(\mathbf{x})\| = 0$ and that point satisfies $\nabla^2 f(\mathbf{x}) \succ 0$. Nonconvex functions may have many minima and many other critical points at which $\left\|\nabla f(\mathbf{x}^\dagger)\right\| = 0$, but the Hessian is not positive definite. Of particular significance are strong saddle points, which are defined—when f is twice differentiable and Morse—as those at which the Hessian is indefinite:

$$\|\nabla f(\mathbf{x}^\ddagger)\| = 0, \qquad \nabla^2 f(\mathbf{x}^\ddagger) \not\succ 0, \qquad \nabla^2 f(\mathbf{x}^\ddagger) \not\prec 0. \tag{3}$$

Local minima can be found with local descent methods. The most widely used of these is gradient descent, which can be proven to approach some $\mathbf{x}^\dagger$ with probability one relative to a random initialization under some standard regularity conditions [24, 30]. Convergence guarantees notwithstanding, gradient descent methods can perform poorly around saddle points. Indeed, while escaping saddles is guaranteed in theory, the number of iterations required to do so is large enough that gradient descent can converge to saddles in practical implementations [31].

Newton's method ameliorates slow convergence of gradient descent by premultiplying gradients with the Hessian inverse. Since the Hessian is positive definite for strongly convex functions, Newton's method provides a descent direction and converges to the minimizer at a quadratic rate in a neighborhood of the minimum. The reason for the improvement in the convergence of Newton's method as compared with gradient descent is due to the fact that by premultiplying the descent direction by the inverse of the Hessian we are performing a local change of coordinates by which the level sets of the function become spherical. The algorithm proposed here relies on performing an analogous transformation that turns saddles with "slow" unstable manifolds compared to the stable manifold—that is, smaller absolute value of the negative eigenvalues of the Hessian than its positive eigenvalues—into saddles that have the same absolute values of every eigenvalue. For nonconvex functions the Hessian is not necessarily positive definite, and convergence to a minimum is not guaranteed by choosing as search direction $-\nabla^2 f(\mathbf{x})^{-1}\nabla f(\mathbf{x})$. In fact, all critical points are stable relative to the aforementioned dynamics and the method can converge to a local minimum, a saddle, or a local maximum. This shortcoming can be overcome by adopting a modified inverse using the absolute values of the Hessian eigenvalues [29].

DEFINITION 2.1 (PT-inverse). *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a symmetric matrix, $\mathbf{Q} \in \mathbb{R}^{n\times n}$ a basis of orthonormal eigenvectors of $\mathbf{A}$, and $\mathbf{\Lambda} \in \mathbb{R}^{n\times n}$ a diagonal matrix of corresponding eigenvalues. We say that $|\mathbf{\Lambda}|_m \in \mathbb{R}^{n\times n}$ is the positive definite truncated (PT)-eigenvalue matrix of $\mathbf{A}$ with parameter m if*

$$\left(|\mathbf{\Lambda}|_m\right)_{ii} = \begin{cases} |\mathbf{\Lambda}_{ii}| & \textit{if} \quad |\mathbf{\Lambda}_{ii}| \geq m, \\ m & \textit{otherwise.} \end{cases} \tag{4}$$

The PT-inverse of $\mathbf{A}$ with parameter m is the matrix $|\mathbf{A}|_m^{-1} = \mathbf{Q}\,|\mathbf{\Lambda}|_m^{-1}\,\mathbf{Q}^\top$.

Given the decomposition $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$, the inverse, when it exists, can be written as $\mathbf{A}^{-1} = \mathbf{Q}\mathbf{\Lambda}^{-1}\mathbf{Q}^\top$. The PT-inverse $|\mathbf{A}|_m^{-1} = \mathbf{Q}\,|\mathbf{\Lambda}|_m^{-1}\,\mathbf{Q}^\top$ flips the signs of the negative eigenvalues and truncates small eigenvalues by replacing m for any eigenvalue with absolute value smaller than m. Both of these properties are necessary to obtain a convergent Newton-like method for nonconvex functions. We use the PT-inverse of the Hessian to define the NCN method. To do so, consider iterates $\mathbf{x}_k$ and a step size $\eta_k > 0$, and use the shorthand $\mathbf{H}_m(\mathbf{x}_k)^{-1} = \left|\nabla^2 f(\mathbf{x}_k)\right|_m^{-1}$ to represent the PT-inverse of the Hessian evaluated at the $\mathbf{x}_k$ iterate. The NCN method is defined by the recursion

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \mathbf{H}_m(x_k)^{-1}\nabla f(\mathbf{x}_k) = \mathbf{x}_k - \eta_k \left|\nabla^2 f(\mathbf{x}_k)\right|_m^{-1} \nabla f(\mathbf{x}_k). \tag{5}$$

Algorithm 1 Nonconvex Newton Step.

```
1: function $[\mathbf{x}_{k+1}] = \text{NCNstep}(\mathbf{x}_k, \alpha, \beta)$
2: Compute PT-inverse: $\mathbf{H}_m(\mathbf{x}_k)^{-1} = \left|\nabla^2 f(\mathbf{x}_k)\right|_m^{-1}$.
3: Set step size to $\eta_k = 1$. Update argument to $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \mathbf{H}_m(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k)$.
4: while $f(\mathbf{x}_{k+1}) > f(\mathbf{x}_k) - \alpha\eta_k \nabla f(\mathbf{x}_k)^\top \mathbf{H}_m(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k)$ do {Backtracking}
5:     Reduce step size to $\eta_k = \beta\eta_k$.
6:     Update argument to $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \mathbf{H}_m(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k)$.
7: end while {return $\mathbf{x}_{k+1}$}
```

The step size η_k is chosen with a backtracking line search as is customary in the regular Newton's method; see, e.g., [3, section 9.5.2]. This yields a step routine that is summarized in Algorithm 1. In step 3 we update the iterate $\mathbf{x}_k$ using the PT-inverse Hessian $\mathbf{H}_m(\mathbf{x}_k)^{-1}$ computed in step 2 and initial step size $\eta_k = 1$. The updated variable $\mathbf{x}_{k+1}$ is checked against the decrement condition with parameter $\alpha \in (0, 0.5)$ in step 4. If the condition is not met, we decrease the step size η_k by backtracking it with the constant $\beta < 1$ as in step 5. We update the iterate $\mathbf{x}_k$ with the new step size as in step 6 and repeat the process until the decrement condition is satisfied.

Since the PT-inverse is defined to guarantee that $\mathbf{H}_m(x_k)^{-1}\nabla f(\mathbf{x}_k)$ is a proper descent direction, it is unsurprising that NCN converges to a local minimum. The expectation is, however, that it will do so at a faster rate because of the Newton-like correction that is implied by (5). Intuitively, the Hessian inverse in convex functions implements a change of coordinates that renders level sets approximately spherical around the current iterate $\mathbf{x}_k$. The Hessian PT-inverse in nonconvex functions implements an analogous change of coordinates that renders level sets in the neighborhood of a saddle point close to a symmetric hyperboloid. The latter is expected to improve convergence, something that has been observed empirically [13, 37, 23, 8, 31].

2.1. Convergence of NCN to local minima.

Convergence results are derived with customary assumptions on Lipschitz continuity of the gradient and Hessian, boundedness of the norm of the local minima, and nondegeneracy of critical points.

ASSUMPTION 1. *The function $f(\mathbf{x})$ is twice continuously differentiable. The gradient and Hessian of $f(\mathbf{x})$ are Lipschitz continuous, i.e., there exist constants $M, L > 0$ such that for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$*

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq M\|\mathbf{x} - \mathbf{y}\| \quad \textit{and} \quad \left\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\right\| \leq L\|\mathbf{x} - \mathbf{y}\|. \tag{6}$$

ASSUMPTION 2. *There exists a positive constant B such that $\|\mathbf{x}^\dagger\| \leq B$ for all $\mathbf{x}^\dagger$ satisfying* (2)*. In particular, this is true of the global minimum $\mathbf{x}^*$ in* (1)*.*

ASSUMPTION 3. *Local minima and saddles are nondegenerate. That is, there exists a constant $\xi > 0$ such that $\min_{i=1\dots n} \lambda_i\left(\nabla^2 f(\mathbf{x}^\dagger)\right) > \xi$ for all local minima $\mathbf{x}^\dagger$, and $\min_{i=1\dots n}\left|\lambda_i\left(\nabla^2 f(\mathbf{x}^\ddagger)\right)\right| > \xi$ for all saddle points $\mathbf{x}^\ddagger$ defined in* (3)*. The notation $\lambda_i(\nabla^2 f(\mathbf{x}))$ refers to the ith eigenvalue of the Hessian of f at the point $\mathbf{x}$.*

The main feature of the update in (5) is that it exploits curvature information to accelerate the rate for escaping from saddle points relative to gradient descent. In particular, the iterates of NCN escape from a local neighborhood of saddle points exponentially fast at a rate which is independent of the problem's condition number. This neighborhood is defined as $\|\nabla f(\mathbf{x})\| < \delta/2$, where

$$\delta = \min\left\{\frac{m^2\,(1-2\alpha)}{L}, \frac{m^2}{5L}\right\}. \tag{7}$$

Throughout the paper, we make the assumption that the accuracy with which we want to solve the problem is ε satisfies $\varepsilon < \delta$. To state this result formally, let $\mathbf{x}^\ddagger$ be a saddle of interest, and denote $\mathbf{Q}_-$ and $\mathbf{Q}_+$ as the orthogonal subspaces associated with the negative and positive eigenvalues of $\nabla^2 f(\mathbf{x}^\ddagger)$. For a point $\mathbf{x} \neq \mathbf{x}^\ddagger$ we define the gradient projections on these subspaces as

$$\nabla f_-(\mathbf{x}) := \mathbf{Q}_-^\top \nabla f(\mathbf{x}) \quad \text{and} \quad \nabla f_+(\mathbf{x}) := \mathbf{Q}_+^\top \nabla f(\mathbf{x}). \tag{8}$$

These projections have different behaviors in the neighborhood of a saddle point. The projection on the positive subspace $\nabla f_+(\mathbf{x})$ enjoys an approximately quadratic convergent phase as in Newton's method (Theorem 3.2). This is as would be expected because the positive portion of the Hessian is not affected by the PT-inverse. The negative portion $\nabla f_-(\mathbf{x})$ can be shown to present an exponential divergence from the saddle point with a rate independent of the problem conditioning. These results provide a bound on the number of steps required to escape the neighborhood of the saddle point that we state next.

Theorem 2.2. *Let $f(\mathbf{x})$ be a function satisfying Assumptions* 1 *and* 3*, let $\varepsilon > 0$ be the desired accuracy of the solution provided by Algorithm* 2*, and let $\alpha \in (0,1)$ be one of its inputs. If $m < \xi/2$,*

$$\|\nabla f_-(\mathbf{x}_0)\| \geq \max\{(5L/2m^2)\,\|\nabla f(\mathbf{x}_0)\|^2, \varepsilon\}, \tag{9}$$

and $\|\nabla f(\mathbf{x}_0)\| \leq \delta/2$, we have that $\|\nabla f(\mathbf{x}_{K_1})\| \geq \delta/2$, with

$$K_1 \leq 1 + \log_{3/2}\left(\frac{\delta}{2\varepsilon}\right). \tag{10}$$

Proof. This theorem is a particular case of Theorem 3.2 in section 3. ∎

The result in Theorem 2.2 establishes an upper bound for the number of iterations needed to escape the saddle point which is of the order $O(log(1/\varepsilon))$ as long as the iterate $\mathbf{x}_0$ satisfies $\|\nabla f_-(\mathbf{x}_0)\| \geq ((5L)/(2m^2))\,\|\nabla f(\mathbf{x}_0)\|^2$ and $\|\nabla f_-(\mathbf{x}_0)\| \geq \varepsilon$. However, the fundamental result is that the rate at which the iterates escape the neighborhood of the saddle point is a constant $3/2$ independent of the specific problem. To establish convergence to a local minimum we will prove four additional results:

(i) In Proposition 3.7 we state that convergence to a neighborhood of the critical points such that $\|\nabla f(\mathbf{x}_k)\| < \delta/2$ is achieved in a constant number of iterations bounded by

$$K_2 = \frac{4M^2\,(f(\mathbf{x}_0) - f(\mathbf{x}^*))}{\alpha\beta m\delta^2}. \tag{11}$$

(ii) We show in Proposition 3.8 that the number of times that the iterates revisit the same neighborhood of a saddle point is upper bounded by

$$T < \frac{2}{\alpha\beta}\frac{M^3}{m^3} + \alpha\beta, \tag{12}$$

and that

(iii) once in such a neighborhood of a local minimum, the algorithm achieves ε accuracy in a number of iterations bounded by (Corollary 3.3)

$$K_3 = \log_2\left(\log_2\left(\frac{2m^2}{5L\varepsilon}\right)\right). \tag{13}$$

(iv) For the case that the iterate is within the neighborhood of a saddle point, but the conditions required by Theorem 2.2 are not satisfied, we show that by adding noise to the iterate we can ensure that said conditions are met with probability $(1-p)^{1/S}$ after a number of iterations of order $O(\log(\log(1/\varepsilon)) + \log(S/p))$, where S is the maximum number of saddles that the algorithm visits. To converge to the minimum with probability $1-p$ we need to escape each of them with probability $(1-p)^{1/S}$. In particular, we show that if we add a bounded version of the Gaussian noise $\mathcal{N}(0, 2\varepsilon/m)$ to each component of the decision variable, with probability q the perturbed variable will be in the region where conditions required by Theorem 2.2 are satisfied. We further show that the probability q is lower bounded by

$$q > 2\left(1-\Phi(1)\right)\frac{\gamma\left(\frac{n}{2},\frac{n}{2}\right)}{\Gamma\left(\frac{n}{2}\right)}, \tag{14}$$

where $\Phi(1)$ is the integral of the Gaussian distribution with integration boundaries $-\infty$ and 1, $\Gamma(\cdot)$ is the gamma function, $\gamma(\cdot,\cdot)$ is the lower incomplete gamma function and n is the dimension of the iterate $\mathbf{x}$. In practice, however, we cannot verify whether the conditions required in Theorem 2.2 are satisfied by the perturbed iterate. To solve this issue, we show that the probability q defined in (14) is in fact the probability of falling in the region $\|\nabla f_-(\mathbf{x}_0)\| \geq \max\{(5L/2m^2)\|\nabla f(\mathbf{x}_0)\|^2, \varepsilon/2\}$. In this region, the iterates present the same behavior as in Theorem 2.2, and therefore the number of steps required to escape the ε neighborhood is $1 + \log_{3/2}(2\varepsilon/\varepsilon) < 2$. Hence, after the perturbation, we run the algorithm for two iterations, i.e., the maximum number of required iterations to escape the ε neighborhood of the saddle assuming that the perturbed iterate is in the preferable region. If the perturbed variable were not in the region of interest, the iterates may not escape the saddle and another round of perturbation is needed. Using this scheme we show that the iterates will be within the desired region of the saddle with probability $(1-p)^{1/S}$ after at most

$$K_4 \leq \left(1 + \frac{\log(S/p)}{\log\left(1/(1-q)\right)}\right)\left[\log_2\left(\log_2\left(\frac{5L}{2m^2\varepsilon}\right)\right) + \log_{3/2}(2) + 1\right]. \tag{15}$$

The fact that we may need to visit each saddle T times does not affect the previous probability, since the neighborhood in which noise is added is reached in only one of these visits (cf. Proposition 3.8). By combining the previous bounds, the total complexity is of order $O(\log(1/p) + \log(1/\varepsilon))$ for NCN to converge to an ε neighborhood of a local minimum with probability $1-p$. We formalize this result next.

Theorem 2.3. *Let $f(\mathbf{x})$ be a function satisfying Assumptions* 1–3. *Let $\varepsilon > 0$ be the accuracy of the solution and $\alpha \in (0, 1/2)$, $\beta \in (0,1)$ the remaining inputs of Algorithm* 2. *Then if $m < \xi/2$, with probability $1-p$ and for any ε satisfying*

$$\varepsilon < \min\left\{\frac{1}{8Ln}, \frac{\delta m}{4M\sqrt{n}+m}, \frac{L}{5\left(2M\sqrt{n}+m\right)^2}, \sqrt{\frac{\alpha\beta\frac{m^3}{M^3}(\delta/2)^2}{1+\left(2\sqrt{n}\frac{M}{m}+1\right)^2}}\right\}, \tag{16}$$

Algorithm 2 Nonconvex Newton Method.

```
1:  Input: $\mathbf{x}_k = \mathbf{x}_0$, accuracy $\varepsilon > 0$, and parameters $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$
2:  while $\|\nabla f(\mathbf{x}_k)\| > \varepsilon$ or $\min \lambda(\nabla^2 f(\mathbf{x}_k)) < 0$ do
3:    Compute the updated variable $\mathbf{x}_{k+1} = \text{NCNstep}(\mathbf{x}_k, \alpha, \beta)$.
4:    if $\|\nabla f(\mathbf{x}_{k+1})\| \leq \varepsilon$ and $\min \lambda(\nabla^2 f(\mathbf{x}_{k+1})) < 0$ then
5:      $\tilde{\mathbf{x}} = \mathbf{x}_{k+1} + X$ with $X_i \sim \mathcal{N}(0, 2\varepsilon/m)$
6:      while $\|\nabla f(\tilde{\mathbf{x}})\| > (2\sqrt{n}M/m + 1)\varepsilon$ do
7:        $\tilde{\mathbf{x}} = \mathbf{x}_{k+1} + X$ with $X_i \sim \mathcal{N}(0, 2\varepsilon/m)$
8:      end while
9:      $\mathbf{x}_{k+1} = \tilde{\mathbf{x}}$
10:     if $\|\nabla f(\mathbf{x}_{k+1})\| \leq \varepsilon$ then
11:       Update the iterate $\mathbf{x}_{k+1} = \text{NCNstep}(\mathbf{x}_k, \alpha, \beta)$ for $\lceil \log_{3/2} 2 \rceil = 2$ times.
12:     end if
13:   end if
14: end while
```

Algorithm 2 *outputs* $\mathbf{x}_{sol}$ *satisfying* $\|\nabla f(\mathbf{x}_{sol})\| < \varepsilon$ *and* $\nabla^2 f(\mathbf{x}_{sol}) \succ 0$ *in at most*

$$K = STK_1 + (ST + 1)K_2 + K_3 + SK_4 \tag{17}$$

iterations, where K_1*,* K_2*,* K_3*,* K_4*, and* T *are the bounds are defined in* (10)–(15) *and* S*—the maximum number of saddles that the algorithm approaches—is bounded by*

$$S \leq \frac{m^2}{M\alpha\beta \left(\delta/2\right)^2} \left(f(\mathbf{x}_0) - f(\mathbf{x}^*)\right). \tag{18}$$

The result in Theorem 2.3 states that the proposed NCN method outputs a solution with a desired accuracy and with a desired probability in a number of steps that is bounded by K. The final bound follows from the fact that it may be required to visit every saddle point T times before reaching a local minimum. Hence we may have $TS + 1$ approaches to neighborhoods of the critical points, each one of them taking K_2 iterations. The latter corresponds to the second term in (17). The first term corresponds to the need to escape all S saddles, T times each if we are in the good region, thus taking a total of STK_1 steps. If we are not in the desired region, it takes K_4 steps to reach said region and we may have to do this in all S saddles, hence the last term in (17). Finally, term K_3 corresponds to the quadratic convergence to the local minimum. The dominant terms in (17) are K_1, which has a $O(\log(1/\varepsilon))$ dependence, and K_4, which depends on the probability as $O(\log(1/p))$.

Before proving the result of the previous theorem we describe the details of Algorithm 2. Its main core is the NCN step described in (5) (step 3) that is performed as long as the iterates are not in a neighborhood of a local minimum satisfying $\|\nabla f(\mathbf{x}_k)\| < \varepsilon$. Steps 4–12 are introduced to add Gaussian noise to satisfy the hypothesis of Theorem 2.2. If the iterate is in a neighborhood of a saddle point such that $\|\nabla f(\mathbf{x}_k)\| < \varepsilon$ (step 4), noise from a Gaussian distribution is added (step 5). The draw is repeated as long as $\|\nabla f(\mathbf{x}_k)\| > (2\sqrt{n}M/m+1)\varepsilon$ (steps 6–8). This is done to ensure that the iterates remain close to the saddle point. Once this condition is satisfied we perform the NCN step twice if the iterate is still in the neighborhood $\|\nabla f(\mathbf{x}_k)\| < \varepsilon$ (steps 10–12). In cases where $\|\nabla f(\mathbf{x}_k)\| \geq (5L/2m^2)\, \|\nabla f(\mathbf{x}_k)\|^2$, two steps of NCN are enough to escape the ε neighborhood of the saddle point and therefore to satisfy the hypothesis of Theorem 2.2 (cf. Proposition 3.6).

2.2. An illustrative example. In understanding escape from a saddle point it is important to distinguish challenges associated with the saddle's condition number and challenges associated with starting from an initial point that is close to the stable manifold. To illustrate these two different challenges we consider a family f_λ of nonconvex functions parametrized by a coefficient $\lambda \in (0, 1]$ and defined as

$$f_\lambda(\mathbf{x}) \;=\; \frac{1}{2}x_1^2 - \frac{\lambda}{2}x_2^2. \tag{19}$$

As λ decreases from 1 to 0 the saddle becomes flatter, its condition number λ^{-1} growing from 1 to $+\infty$. Gradient descent iterates using unit step size for the function f_λ result in zeroing of the first coordinate in just one step because $x_1^+ = x_1 - \eta\nabla_{x_1} f_\lambda(\mathbf{x}) = x_1 - x_1 = 0$. The second coordinate evolves according to the recursion

$$x_2^+ = x_2 - \eta\nabla_{x_2} f_\lambda(\mathbf{x}) = (1+\lambda)x_2. \tag{20}$$

Likewise, NCN with unit step size results in zeroing of the first coordinate because $x_1^+ = x_1 - \eta[\mathbf{H}^{-1}\nabla f_\lambda(\mathbf{x})]_1 = 0$. The second coordinate, however, evolves as

$$x_2^+ = x_2 - \left[\mathbf{H}^{-1}\nabla f_\lambda(\mathbf{x})\right]_2 = 2x_2. \tag{21}$$

Both expressions imply exponential escape from the saddle point. In the case of gradient descent the base of escape is $(1 + \lambda)$, but in the case of NCN the base of escape is 2, independent of the value of λ.

The consequences of this difference are illustrated in Figure 1 which depicts the number of iterations it takes to reach the border of the L_∞ unit ball as a function of the condition number λ^{-1}. Different curves depict different initial conditions in terms of their distance γ to the stable manifold $x_2 = 0$, which is simply the initial value for the x_2 coordinate. The escape from the saddle for NCN occurs in $\ln\gamma/\ln 2$ iterations, a number that is independent of the condition number. The time it takes for gradient descent to escape a saddle is $\ln\gamma/\ln(1+\lambda) \approx \lambda^{-1}\ln\gamma$. This number is roughly proportional to the saddle's condition number. An important observation that follows from Figure 1 is that the condition number of the saddle is a more challenging problem than the initial distance to the stable manifold. If the saddle is well-conditioned, escape from the saddle with gradient descent takes a few iterations even when the initial condition is very close to the stable manifold. For example, $\ln\gamma/\ln 2 < 67$ when $\gamma = 10^{-20}$. If the saddle is not well-conditioned, escape from the saddle with gradient descent takes a very large number of iterations even if the initial condition is far from the stable manifold. For example, $\ln\gamma/\ln(1+\lambda) > 10^5$ when $\gamma = 0.1$ and $\lambda = 10^{-5}$. This is because the number of iterations to escape a saddle is approximately $\lambda^{-1}\ln\gamma$. This number grows logarithmically with γ but linearly with the condition number λ^{-1}. In the case of NCN, escape always takes $\ln\gamma/\ln 2$ iterations independently of λ. Theorem 2.2 provides a qualitatively analogous statement for any saddle.

3. Convergence analysis.

To study the convergence of the proposed NCN method we divide the results into two parts. In the first part, we study the performance of NCN in a neighborhood of critical points. We first define this region and

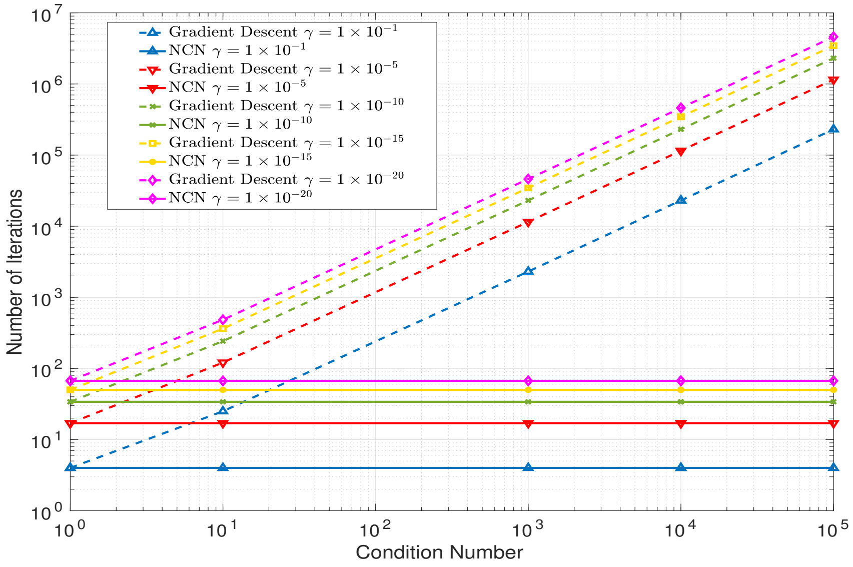

FIG. 1. *Number of iterations to escape the neighborhood* $[-1, 1]^2$ *for quadratic problems with different condition numbers* λ^{-1} *and different initial distance to the stable manifold* γ. *NCN takes the same number of steps to escape the neighborhood of the saddle point regardless of the condition number of the problem. In the case of gradient descent this dependence is linear.*

then characterize the number of required iterations to escape it in the case that the critical point is a saddle, or the number of required iterations for convergence in the case that the critical point is a minimum. In the second part, we study the behavior of NCN when the iterates are not close to a critical point and derive an upper bound for the number of iterations required to reach one.

To analyze the local behavior of NCN we characterize the region in which the step size chosen by backtracking line search is $\eta = 1$, as in the standard Newton's method for convex optimization. We formally introduce this region in the following lemma.

LEMMA 3.1. *Let* f *be a function satisfying Assumptions* 1 *and* 3, *and let* $\alpha \in (0, 1/2)$ *be the input parameter of Algorithm* 2. *Then if* $\|\nabla f(\mathbf{x})\| \leq (m^2/L)\,(1 - 2\alpha)$, *a backtracking algorithm admits the step size* $\eta = 1$.

Proof. See Appendix B. □

The result in Lemma 3.1 characterizes the neighborhood in which the step size of NCN chosen by backtracking is $\eta = 1$. In the following theorem, we study the behavior of NCN in this neighborhood. Before introducing the result, we recall the definitions of the gradient projected over the subspace of the eigenvectors associated with the negative and positive eigenvalues in (8) which are denoted by $\nabla f_-(\mathbf{x})$ and $\nabla f_+(\mathbf{x})$, respectively. We attempt to show that the norm $\|\nabla f_-(\mathbf{x}_k)\|$ is almost doubled per iteration in this local neighborhood, and the norm $\|\nabla f_+(\mathbf{x})\|$ converges to zero quadratically.

THEOREM 3.2. *Let Assumptions* 1–3 *and* $\|\nabla f(\mathbf{x}_k)\| < m^2(1 - 2\alpha)/L$ *hold,* $m <$

$\xi/2$. Then $\nabla f_-(\mathbf{x})$ and $\nabla f_+(\mathbf{x})$ defined in (8) *satisfy*

$$\|\nabla f_+(\mathbf{x}_{k+1})\| \leq \frac{5L}{2m^2}\|\nabla f(\mathbf{x}_k)\|^2, \tag{22}$$

$$\|\nabla f_-(\mathbf{x}_{k+1})\| \geq 2\|\nabla f_-(\mathbf{x}_k)\| - \frac{5L}{2m^2}\|\nabla f(\mathbf{x}_k)\|^2. \tag{23}$$

Proof. It follows from Lemma 3.1 that the backtracking algorithm admits a step $\eta = 1$. Hence we can write $\nabla f(\mathbf{x}_{k+1})$ as

$$\nabla f(\mathbf{x}_{k+1}) = \nabla f(\mathbf{x}_k) + \int_0^1 \nabla^2 f(\mathbf{x}_k + \theta\Delta\mathbf{x})\Delta\mathbf{x}d\theta. \tag{24}$$

We next show that in the region $\|\nabla f(\mathbf{x}_k)\| < m^2(1-2\alpha)/L$ it holds that $m < \min_{i=1\dots n}\left|\lambda_i\left(\nabla^2 f(\mathbf{x})\right)\right|$. Let us consider the region $\|\mathbf{x}_k - \mathbf{x}_c\| < m(1-2\alpha)/L$. In this region, by virtue of Lemma A.2 we have that $\min_{i=1\dots n}\left|\lambda_i\left(\nabla^2 f(\mathbf{x})\right)\right| > m$ and that in the boundary $\|\nabla f(\mathbf{x}_k)\| \geq m^2(1-2\alpha)/L$. Since at the critical point $\|\nabla f(\mathbf{x}_c)\| = 0$, by continuity of the norm we have that the region $\|\nabla f(\mathbf{x}_k)\| < m^2(1-2\alpha)/L$ is contained in the region in which $\min_{i=1\dots n}\left|\lambda_i\left(\nabla^2 f(\mathbf{x})\right)\right| > m$. Thus, we have that $\mathbf{H}_m(\mathbf{x}_k) = \mathbf{Q}(\mathbf{x}_k)\left|\Lambda(\mathbf{x}_k)\right|\mathbf{Q}(\mathbf{x}_k)^\top$. Using the fact that $\nabla f(\mathbf{x}_k) = \mathbf{H}_m(\mathbf{x}_k)\mathbf{H}_m(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k) = -\mathbf{H}_m(\mathbf{x}_k)\Delta\mathbf{x}$, the previous expression can be written as

$$\begin{aligned}\nabla f(\mathbf{x}_{k+1}) &= 2\nabla f(\mathbf{x}_k) + \int_0^1 \left(\nabla^2 f(\mathbf{x}_k+\theta\Delta\mathbf{x}) + \mathbf{H}_m(\mathbf{x}_k)\right)\Delta\mathbf{x}d\theta \\ &= 2\nabla f(\mathbf{x}_k) + \int_0^1 \left(\nabla^2 f(\mathbf{x}_k+\theta\Delta\mathbf{x}) - \nabla^2 f(\mathbf{x}_k)\right)\Delta\mathbf{x}d\theta \\ &+ \int_0^1 \left(\nabla^2 f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_c)\right)\Delta\mathbf{x}d\theta + \int_0^1 \left(\nabla^2 f(\mathbf{x}_c) + \mathbf{H}_m(\mathbf{x}_c)\right)\Delta\mathbf{x}d\theta \\ &+ \int_0^1 \left(\mathbf{H}_m(\mathbf{x}_k) - \mathbf{H}_m(\mathbf{x}_c)\right)\Delta\mathbf{x}d\theta.\end{aligned} \tag{25}$$

The latter equality follows by adding and subtracting $\nabla^2 f(\mathbf{x}_c)\Delta\mathbf{x}$, $\nabla^2 f(\mathbf{x}_k)\Delta\mathbf{x}$, and $\mathbf{H}_m(\mathbf{x}_c)\Delta\mathbf{x}$. Multiply both sides of (25) by $\mathbf{Q}_-^\top$, the matrix of the eigenvectors corresponding to negative eigenvalues of the Hessian at $\mathbf{x}_c$. Since $\mathbf{Q}_-$ is a matrix whose columns are eigenvectors, its norm is bounded by one. Combine this fact with the Lipschitz continuity of the Hessian (cf. Assumption 1) to write

$$\left\|\mathbf{Q}_-^\top\left(\nabla^2 f(\mathbf{x}_k+\theta\Delta\mathbf{x}) - \nabla^2 f(\mathbf{x}_k)\right)\Delta\mathbf{x}\right\| \leq L\theta\|\Delta\mathbf{x}\|^2. \tag{26}$$

Likewise, we can upper bound the second and fourth integrands in (25) by

$$\left\|\mathbf{Q}_-^\top\left(\nabla^2 f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_c)\right)\Delta\mathbf{x}\right\| \leq L\|\mathbf{x}_k - \mathbf{x}_c\|\|\Delta\mathbf{x}\|, \tag{27}$$

$$\left\|\mathbf{Q}_-^\top\left(\mathbf{H}_m(\mathbf{x}_k) - \mathbf{H}_m(\mathbf{x}_c)\right)\Delta\mathbf{x}\right\| \leq L\|\mathbf{x}_k - \mathbf{x}_c\|\|\Delta\mathbf{x}\|. \tag{28}$$

We next show that the third integrand in (25) when multiplied by $\mathbf{Q}_-^\top$ becomes zero. Let us write the product $\mathbf{Q}_-^\top\left(\nabla^2 f(\mathbf{x}_c) + \mathbf{H}_m(\mathbf{x}_c)\right)$ as

$$\mathbf{Q}_-^\top\left(\nabla^2 f(\mathbf{x}_c) + \mathbf{H}_m(\mathbf{x}_c)\right) = \mathbf{Q}_-^\top\mathbf{Q}\left(\mathbf{\Lambda}(\mathbf{x}_c) + |\mathbf{\Lambda}(\mathbf{x}_c)|\right)\mathbf{Q}^\top. \tag{29}$$

Let d be the number of negative eigenvalues. Then $\mathbf{Q}_-^\top \mathbf{Q} = [\mathbf{I}_d, \mathbf{0}_{d\times n-d}]$. Moreover, $\boldsymbol{\Lambda}(\mathbf{x}_c) + |\boldsymbol{\Lambda}(\mathbf{x}_c)|$ is diagonal with the first d elements being zero and the remaining $n-d$ being 2. This shows that $\mathbf{Q}_-^\top \mathbf{Q}\left(\boldsymbol{\Lambda}(\mathbf{x}_c) + |\boldsymbol{\Lambda}(\mathbf{x}_c)|\right) = \mathbf{0}$. With this result and the bounds on (26)–(28) we can lower bound (25) by

$$\|\nabla f_-(\mathbf{x}_{k+1})\| \geq 2\|\nabla f_-(\mathbf{x}_k)\| - L\|\Delta\mathbf{x}\|^2 \int_0^1 \theta d\theta - 2L\|\mathbf{x}_k - \mathbf{x}_c\|\|\Delta\mathbf{x}\|. \tag{30}$$

Finally, using the fact that $\|\nabla f(\mathbf{x}_k)\| \geq m\|\mathbf{x}_k - \mathbf{x}_c\|$ (cf. Lemma A.2) and that $\|\Delta\mathbf{x}\| \leq \frac{1}{m}\|\nabla f(\mathbf{x}_k)\|$, the previous bound reduces to

$$\|\nabla f_-(\mathbf{x}_{k+1})\| \geq 2\|\nabla f_-(\mathbf{x}_k)\| - \frac{5L}{2m^2}\|\nabla f(\mathbf{x}_k)\|^2. \tag{31}$$

The proof for the projection over the positive subspace is analogous. □

The first result in Theorem 3.2 shows that the norm $\|\nabla f_+(\mathbf{x}_{k+1})\|$ approaches zero at a quadratic rate if most of the energy of the gradient norm $\|\nabla f(\mathbf{x}_k)\|$ belongs to the term $\|\nabla f_+(\mathbf{x}_k)\|$. In particular, when we are in a local neighborhood of a local minimizer and all the eigenvalues are positive we have $\|\nabla f(\mathbf{x}_k)\| = \|\nabla f_+(\mathbf{x}_k)\|$, and therefore the sequence of iterates converges quadratically to the local minimum. Indeed, in a neighborhood of a local minimum the algorithm proposed here is equivalent to Newton's method. We formalize this result in the following corollary.

Corollary 3.3. *Let f be a function satisfying Assumptions* 1 *and* 3*, and let $\mathbf{x}_0 \in \mathbb{R}^n$ be in the neighborhood of a local minimum such that $\|\nabla f(\mathbf{x}_0)\| < \delta$, where δ is defined as in* (7) *and $m < \xi/2$. Then, it holds that $\|\nabla f(\mathbf{x}_{K_3})\| < \varepsilon$, where K_3 satisfies*

$$K_3 \leq \log_2\left(\log_2\left(\frac{2m^2}{5L\varepsilon}\right)\right). \tag{32}$$

Proof. See Appendix C. □

The second result in Theorem 3.2 shows that the norm $\|\nabla f_-(\mathbf{x}_k)\|$ multiplies by factor $2-\zeta$, where $0 < \zeta < 1$ is a free parameter, after each iteration of NCN if the squared norm $\|\nabla f(\mathbf{x}_k)\|^2$ is negligible relative to $\|\nabla f_-(\mathbf{x}_k)\|$. We formally state this result in the following proposition.

Proposition 3.4. *Let f be a function satisfying Assumptions* 1 *and* 3*, let δ be the constant defined in* (7)*, and let $\zeta \in (0,1)$ and $\varepsilon > 0$ be the inputs of Algorithm* 2*. Then, if the conditions $\|\nabla f_-(\mathbf{x}_0)\| \geq \max\{(5L)/(2m^2)\|\nabla f(\mathbf{x}_0)\|^2, \varepsilon\}$ and $\|\nabla f(\mathbf{x}_0)\| \leq \zeta\delta$ hold, with $m < \xi/2$, the sequence generated by NCN needs K_1 iterations to escape the saddle and satisfy the condition $\|\nabla f(\mathbf{x}_{K_1})\| \geq \zeta\delta$, where K_1 is upper bounded by*

$$K_1 \leq 1 + \frac{\log(\zeta\delta/\varepsilon)}{\log(2-\zeta)}. \tag{33}$$

Proof. See Appendix D. □

The result in the previous proposition states that when the norm of the projection of the gradient over the subspace of eigenvectors corresponding to negative eigenvalues $\|\nabla f_-(\mathbf{x}_0)\|$ is larger than the required accuracy ϵ and the expression $(5L)/(2m^2)\|\nabla f(\mathbf{x}_0)\|^2$, then the sequence of iterates generated by NCN escapes the saddle exponentially with base $(2-\zeta)$. Since ζ is a free parameter, we set $\zeta = 1/2$

as in Theorem 2.2. When the condition $\|\nabla f_-(\mathbf{x}_0)\| \geq (5L)/(2m^2)\|\nabla f(\mathbf{x}_0)\|^2$ in Proposition 3.4 is not met for all k, the sequence generated by NCN reaches the ε neighborhood after $\log_2\left(\log_2\left((5L\zeta)/(m^2\varepsilon)\right)\right)$ iterations, as shown in the following proposition.

Proposition 3.5. *Let f be a function satisfying Assumptions 1 and 3. Further, recall the definition of δ in (7) and let ε be the accuracy of Algorithm 2. Then, if $m < \xi/2$, the condition $\|\nabla f_-(\mathbf{x}_0)\| \geq (5L)/(2m^2)\|\nabla f(\mathbf{x}_0)\|^2$ is violated, and $\|\nabla f(\mathbf{x}_0)\| \leq \zeta\delta$ holds for some $\zeta \in (0,1)$, the iterates either satisfy $\|\nabla f(\mathbf{x}_{\tilde{k}})\| \leq \varepsilon$ with*

$$\tilde{k} = \log_2\left(\log_2\left(\frac{5L\zeta}{m^2\varepsilon}\right)\right) \tag{34}$$

or for some $k < \tilde{k}$ we have that $\|\nabla f_-(\mathbf{x}_0)\| \leq (5L)/(2m^2)\|\nabla f(\mathbf{x}_0)\|^2$.

Proof. See Appendix E. □

If the iterate $\mathbf{x}_0$ is in the ε neighborhood of a saddle point, then Algorithm 2 adds uniform noise in each component $X_i \sim \mathcal{N}(0, 2\varepsilon/m)$ to $\mathbf{x}_0$. To analyze the perturbed iteration we define the following set:

$$\mathcal{G} := \left\{\mathbf{x} \in \mathbb{R}^n \middle| \, \|\nabla f_-(\mathbf{x})\| \geq \max\left\{\frac{5L}{2m^2}\|\nabla f(\mathbf{x})\|^2, \gamma\varepsilon\right\}, \|\nabla f(\mathbf{x})\| \leq \zeta\delta\right\}. \tag{35}$$

We show in Lemma F.1 that the probability of $\mathbf{x}_0 + X \in \mathcal{G}$ is lower bounded by q given in (14). In this case NCN escapes the ε neighborhood of the saddle at an exponential rate in $1 + (\log(1/\gamma))/(\log(2-\zeta))$ iterations based on the analysis of Proposition 3.4. If the latter is not the case, we show that the number of iterations between re-sample instances is bounded by $O(\log(\log(1/\varepsilon)))$. Because we want to escape each saddle point with probability $(1-p)^{1/S}$, the number of draws needed is of the order of $\log(S/p)$. The following proposition formalizes the previous discussion.

Proposition 3.6. *Let $f(\mathbf{x})$ be a function satisfying Assumptions 1, 2, and 3. Let $m < \xi/2$ and $\varepsilon > 0$, the desired accuracy of the output of Algorithm 2, be such that*

$$\varepsilon < \min\left\{\frac{1-\gamma}{4Ln}, \frac{\zeta\delta m}{2M\sqrt{n}+m}, \frac{2L\gamma}{5\left(2M\sqrt{n}+m\right)^2}\right\}. \tag{36}$$

Further, consider the constants $\gamma, \zeta \in (0,1)$, and let q be a constant such that

$$q > 2\left(1-\Phi(1)\right)\frac{\gamma\left(\frac{n}{2},\frac{n}{2}\right)}{\Gamma\left(\frac{n}{2}\right)}, \tag{37}$$

where $\Phi(1)$ is the integral of the Gaussian distribution with integration boundaries $-\infty$ and 1, $\Gamma(\cdot)$ is the gamma function, and $\gamma(\cdot,\cdot)$ is the lower incomplete gamma function. For any $\mathbf{x}_0$ in the neighborhood of a saddle point satisfying $\|\nabla f(\mathbf{x}_0)\| \leq \varepsilon$, with probability $(1-p)^{1/S}$ we have that $\|\nabla f_-(\mathbf{x}_{K_4})\| \geq \frac{5L}{2m^2}\|\nabla f(\mathbf{x}_{K_4})\|^2$ and $\varepsilon \leq \|\nabla f_-(\mathbf{x}_{K_4})\|$, where K_4 is given by

$$K_4 = \left(1 + \frac{\log(S/p)}{\log\left(1/(1-q)\right)}\right)\left[\log_2\left(\log_2\left(\frac{5L\zeta}{m^2\varepsilon}\right)\right) + \frac{\log\left(1/\gamma\right)}{\log(2-\zeta)} + 1\right]. \tag{38}$$

Proof. See Appendix F. □

Combining the results from Propositions 3.4 and 3.6, we show that with probability $1-p$ the number of steps required to escape the $\zeta\delta$ neighborhood of a saddle is $O\left(\log(1/p)+\log(1/\varepsilon)\right)$. The previous result completes the analysis of the neighborhoods of the critical points. To complete the convergence analysis we show in the following lemma that the number of iterations required to reach a neighborhood of a critical point satisfying $\|\nabla f(\mathbf{x})\| < \zeta\delta$ is constant.

PROPOSITION 3.7. *Let f be a function satisfying Assumptions* 1–3 *and let $\alpha \in (0,1/2)$, $\beta \in (0,1)$, and $m < \xi/2$ be the inputs of Algorithm* 2. *Further, recall the definition of δ in* (7). *Then if $\|\nabla f(\mathbf{x}_0)\| \geq \zeta\delta$, in at most K_2 iterations, Algorithm* 2 *reaches a neighborhood such that $\|\nabla f(\mathbf{x}_{K_2})\| < \zeta\delta$, with*

$$K_2 \leq \frac{M^2\left(f(\mathbf{x}_0)-f(\mathbf{x}^*)\right)}{\alpha\beta m(\zeta\delta)^2}. \tag{39}$$

Proof. See Appendix G. □

The previous result establishes a bound on the number of iterations that NCN takes to reach a neighborhood of a critical point satisfying $\|\nabla f(\mathbf{x})\| < \zeta\delta$. However, to complete the proof of the complexity of the algorithm, we need to ensure that the algorithm does not visit the neighborhood of the same saddle point indefinitely. In particular the next result establishes that NCN visits such neighborhoods at most a constant number of times and that in at most one of these visits noise is added.

PROPOSITION 3.8. *Let $f(\mathbf{x})$ satisfy Assumptions* 1–3, *and consider $\alpha \in (0,1/2)$, $\beta \in (0,1)$, and $m < \xi/2$ as the inputs of Algorithm* 2. *Let $\mathbf{x}_c$ be a critical point of $f(\mathbf{x})$. Let $\zeta \in (0,1)$, let δ be the constant defined as in Theorem* 2.2, *and define*

$$\mathcal{N} = \left\{\mathbf{x}\in\mathbb{R}^n \middle| \|\mathbf{x}-\mathbf{x}_c\| \leq m(1-2\alpha)/L, \|\nabla f(\mathbf{x})\| < \zeta\delta\right\}. \tag{40}$$

Let $\mathbf{x}_0 \in \mathcal{N}$ and the desired accuracy ε satisfy

$$\varepsilon < \sqrt{\frac{\alpha\beta\frac{m^3}{M^3}(\zeta\delta)^2}{1+\left(2\sqrt{n}\frac{M}{m}+1\right)^2}}. \tag{41}$$

Let $T = \sum_{k=1}^{\infty} \mathbb{1}\left(\mathbf{x}_k \in \mathcal{N}, \mathbf{x}_{k-1} \notin \mathcal{N}\right)$ be the number of times that the sequence generated by NCN visits $\mathcal{N}$. Then, the sequence generated by NCN is such that

$$T < \frac{2}{\alpha\beta}\frac{M^3}{m^3}+\alpha\beta, \tag{42}$$

and the neighborhood $\|\nabla f(\mathbf{x})\| < \varepsilon$ is visited at most once. Likewise, the number of saddle points visited before converging to a local minimum S is upper bounded by

$$S \leq \frac{m^2}{M\alpha\beta\left(\zeta\delta\right)^2}\left(f(\mathbf{x}_0)-f(\mathbf{x}^*)\right). \tag{43}$$

Proof. See Appendix H. □

The proof of the final complexity stated in Theorem 2.3 follows from Propositions 3.4, 3.6, 3.7, 3.8, Corollary 3.3, and the discussion after the theorem in section 2.1.

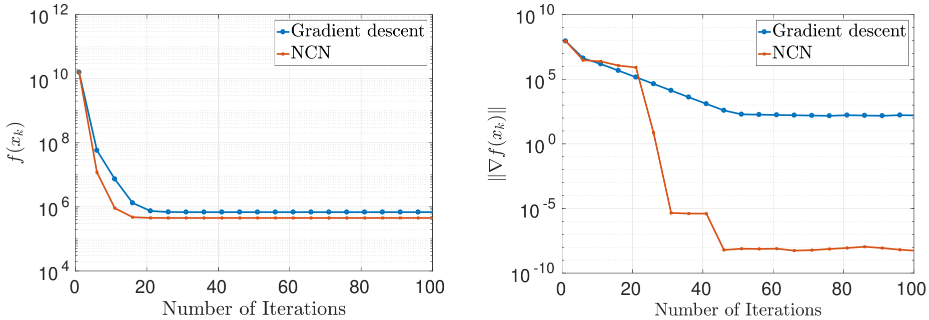

FIG. 2. *Comparison of gradient descent and NCN with $m = 1 \times 10^{-9}$ in terms of value of the objective function (left) and norm of the gradient (right) for the rank $r = 2$ factorization problem. The dimension of the matrix to factorize is* 943×1682.

4. Numerical experiments.

In this section we apply Algorithm 2 to matrix factorization, where the goal is to find a rank r approximation of a matrix $\mathbf{M} \in \mathcal{M}^{l \times n}$, representing ratings from $l = 943$ users to $n = 1682$ movies [19]. Each user has rated at least 20 movies for a total of 99900 known ratings. The problem can be written as

$$(\mathbf{U}^*, \mathbf{V}^*) := \operatorname*{argmin}_{\mathbf{U} \in \mathcal{M}^{l \times r}, \mathbf{V} \in \mathcal{M}^{n \times r}} f(\mathbf{U}, \mathbf{V}) = \operatorname*{argmin}_{\mathbf{U} \in \mathcal{M}^{l \times r}, \mathbf{V} \in \mathcal{M}^{n \times r}} \frac{1}{2} \left\| \mathbf{M} - \mathbf{U}\mathbf{V}^\top \right\|_F^2 . \tag{44}$$

Write the product $\mathbf{U}\mathbf{V}^\top$ as $\sum_{i=1}^r \mathbf{u}_i \mathbf{v}_i^\top$, where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the ith columns of the matrices $\mathbf{U}$ and $\mathbf{V}$, respectively. Let $\mathbf{x} = [\mathbf{u}_1^\top, \ldots, \mathbf{u}_l^\top, \mathbf{v}_1^\top, \ldots \mathbf{v}_n^\top]^\top$; then we solve

$$\mathbf{x}^* := \operatorname*{argmin}_{\mathbf{x} \in \mathbf{R}^{r \times (n+l)}} f(\mathbf{x}). \tag{45}$$

We compare the performance of gradient descent and NCN (Algorithm 2) on the problem (45). The step size in gradient descent is selected via line search by backtracking with parameters α and β being the same as those of NCN. The parameters selected are $r = 2$, $\alpha = 0.1$, $\beta = 0.9$, accuracy $\varepsilon = 1 \times 10^{-8}$, and $m = 1 \times 10^{-12}$. The initial iterate selected for all simulations is the same, and it is selected at random from a multivariate normal random variable with mean zero and standard deviation 10.

In Figure 2 we plot the value of the objective function and the norm of the gradient in logarithmic scale for gradient descent and NCN. After 20 iterations the value of the objective function achieved by NCN is half of that achieved by gradient descent. Notice that the norm of the gradient of the objective function remains large for gradient descent. The latter is an indication of the algorithm being stuck in a neighborhood of a saddle point. Recall from the example in section 2.2 that escaping an ill-conditioned saddle point can take an extremely large number of iterations; hence the fact that the norm seems constant is unsurprising. The minimum eigenvalue of the Hessian of the point to which gradient descent converges is -12.1592, which indicates that it is converging to a neighborhood of a saddle point. On the other hand, the iterates of NCN achieve a point whose minimum eigenvalue of the Hessian is -3.0679×10^{-7}. Even if the latter critical point is formally a saddle, in practice this point can be considered a local minimum with eigenvalue 0. Finally, observe that NCN enjoys a fast convergence in the neighborhood of the local minimum, which is to be expected from Corollary 3.3.

5. Conclusion.

The algorithm presented in this work achieves convergence to a local minimum of a nonconvex function with probability $1-p$ in $O\left(\log(1/p), \log(1/\varepsilon)\right)$

iterations. The main feature of NCN is the curvature correction of the function by premultiplying the gradient by the PT-inverse of the Hessian. Since the latter is a positive definite matrix, it ensures that the NCN is a descent direction for the objective function and therefore it converges to a neighborhood of the critical points. This feature could be achieved by any other positive definite matrix; however, the structure of PT-inverse enforces a special behavior in the neighborhood of the following critical points: (i) The projection of the gradient $\nabla f_+(\mathbf{x}_k)$ exhibits a quadratic convergence behavior. In particular, in the neighborhood of the local minima, this subspace is the whole space, and therefore NCN converges to an ε neighborhood of the local minimum in $O(\log_2(\log_2(1/\varepsilon)))$, as in the case for the classic Newton's method. (ii) The projection of the gradient over the complement of the previous subspace has an exponential escape rate of $3/2$ independent of the conditioning of the problem as long as a considerable part of the energy of the gradient lies in this subspace. When the previous condition is not met, it is possible to add noise, and with probability $1-p$ after $O\log(1/p)$ iterations the iterate satisfies said condition. The randomization was not needed in the numerical examples, since NCN escapes the saddle before reaching its ε neighborhood. This suggests that random initialization is enough to ensure an exponential rate of escape in practice, although this is not supported by theoretical guarantees. The main drawbacks of the proposed algorithm are the need for spectral decomposition, which is impractical in large-scale problems, and the requirement of saddle points to be nondegenerated, which might be a strong assumption for many problems of interest. However, the ability to evade saddle points in a number of iterations that is independent of the condition number is a remarkable property of our algorithm, since the condition number of the saddle has an exponential effect in the number of iterations that other algorithms take to escape the saddle, as illustrated by the example in section 2.2. Hence, this is a promising starting point to develop algorithms with better complexity—computing, for instance, approximations of the PT-inverse, as quasi-Newton methods do to approximate the Hessian inverse—that operate under weaker assumptions.

Appendix A. Consequences of the Lipschitz continuity of the Hessian. In this section we state and prove some useful lemmas needed in section 3.

LEMMA A.1. *Let $f(\mathbf{x})$ satisfy Assumptions* 1 *and* 3*. Then the eigenvalues of the Hessian $\lambda_i(\nabla^2 f(\mathbf{x}))$ are Lipschitz with constant L.*

Proof. We show that for matrices $\mathbf{A},\mathbf{B}\in\mathcal{M}^{n\times n}$, if $\mathbf{A}\preceq\mathbf{B}$, then $\lambda_i(\mathbf{A})\leq\lambda_i(\mathbf{B})$ for all $i=1,\ldots,n$. Note that for a given i, there exists a subspace V_1 of dimension $n-i+1$ where for any $\mathbf{x}\in V_1$ we have $\mathbf{x}^\top\mathbf{A}\mathbf{x}\geq\lambda_i(\mathbf{A})\|\mathbf{x}\|^2$. Likewise, there exists a subspace V_2 of dimension i such that for any $\mathbf{x}\in V_2$ we have $\mathbf{x}^\top\mathbf{B}\mathbf{x}\leq\lambda_i(\mathbf{B})\|\mathbf{x}\|^2$, since the sum of the dimensions of V_1 and V_2 is larger than n. There exists $\mathbf{x}\neq\mathbf{0}$, $\mathbf{x}\in V_1\cap V_2$. Since $\mathbf{A}\preceq\mathbf{B}$ we have that $0\leq\mathbf{x}^\top(\mathbf{B}-\mathbf{A})\mathbf{x}\leq(\lambda_i(\mathbf{B})-\lambda_i(\mathbf{A}))\|\mathbf{x}\|^2$. Therefore $\lambda_i(\mathbf{A})\leq\lambda_i(\mathbf{B})$. Next observe that if $\|\mathbf{A}-\mathbf{B}\|\leq\varepsilon$, it holds that $\varepsilon\mathbf{I}\pm(\mathbf{A}-\mathbf{B})\succeq 0$. Hence, using the previous result we have that $\varepsilon+\lambda_i(\mathbf{A})\geq\lambda_i(\mathbf{B})$ and $\varepsilon+\lambda_i(\mathbf{B})\geq\lambda_i(\mathbf{A})$, which implies that $|\lambda_i(\mathbf{B})-\lambda_i(\mathbf{A})|\leq\varepsilon$. The proof is completed by observing that $\nabla^2 f(\mathbf{x})$ is Lipschitz, and therefore for any $\mathbf{x},\mathbf{y}\in\mathbb{R}^n$ it holds that $\left\|\nabla^2 f(\mathbf{x})-\nabla^2 f(\mathbf{y})\right\|\leq L\|\mathbf{x}-\mathbf{y}\|$. ∎

LEMMA A.2. *Let $f(\mathbf{x})$ satisfy Assumptions* 1 *and* 3*, and let $\mathbf{x}_c$ be a critical point $f(\mathbf{x})$. For any $\mathbf{x}\in\mathbb{R}^n$ such that $\|\mathbf{x}-\mathbf{x}_c\|<m(1-2\alpha)/L$, with $m<\zeta/2$ it holds that*

$$\min_{i=1\ldots n}\lambda_i\left|(\nabla^2 f(\mathbf{x}))\right|\geq m \quad \textit{and} \quad m\|\mathbf{x}-\mathbf{x}_c\|\leq\|\nabla f(\mathbf{x})\|. \tag{46}$$

Proof. Let $\mathbf{x}_c$ be a critical point with $|\lambda_i(\nabla^2 f(\mathbf{x}_c))| > \zeta > 2m$ (cf. Assumption 3) and let $\mathbf{x} \in \mathbb{R}^n$ be such that $\|\mathbf{x}-\mathbf{x}_c\| < m(1-2\alpha)/L$. By the Lipschitz continuity of the eigenvalues (cf. Lemma A.1), for all $i = 1, \ldots, n$ we have that

$$\left|\lambda_i\left(\nabla^2 f(\mathbf{x})\right) - \lambda_i\left(\nabla^2 f(\mathbf{x}_c)\right)\right| \leq L\left\|\mathbf{x}-\mathbf{x}_c\right\| \leq (1-2\alpha)m < m. \tag{47}$$

Since the difference between the ith eigenvalue at $\mathbf{x}$ and $\mathbf{x}_c$ is bounded by m, the first claim in (46) holds. To prove the second claim, write the gradient of $f(\mathbf{x})$ as

$$\nabla f(\mathbf{x}) = \nabla^2 f(\mathbf{x}_c)\left(\mathbf{x}-\mathbf{x}_c\right) + \frac{1}{2}\left(\nabla^2 f(\tilde{\mathbf{x}}) - \nabla^2 f(\mathbf{x}_c)\right)\left(\mathbf{x}-\mathbf{x}_c\right), \tag{48}$$

where $\tilde{\mathbf{x}} = \theta\mathbf{x}_c + (1-\theta)\mathbf{x}$ with $\theta \in [0,1]$. Since the minimum absolute value of the eigenvalues of the Hessian at $\mathbf{x}_c$ is larger than $2m$, we have that

$$\|\nabla f(\mathbf{x})\| \geq 2m\left\|\mathbf{x}-\mathbf{x}_c\right\| - \frac{1}{2}\left\|\left(\nabla^2 f(\tilde{\mathbf{x}}) - \nabla^2 f(\mathbf{x}_c)\right)\left(\mathbf{x}-\mathbf{x}_c\right)\right\|. \tag{49}$$

Using the Lipschitz continuity of the Hessian (cf. Assumption 1), (49) reduces to

$$\begin{aligned}\|\nabla f(\mathbf{x})\| &\geq 2m\left\|\mathbf{x}-\mathbf{x}_c\right\| - \frac{L}{2}\left\|\mathbf{x}-\mathbf{x}_c\right\|^2 \\ &\geq m\left(2 - \frac{L\left\|\mathbf{x}-\mathbf{x}_c\right\|}{2}\right)\left\|\mathbf{x}-\mathbf{x}_c\right\| \geq m\left\|\mathbf{x}-\mathbf{x}_c\right\|,\end{aligned} \tag{50}$$

where the last inequality follows from the fact that $\|\mathbf{x}-\mathbf{x}_c\| < m/L$. □

Appendix B. Proof of Lemma 3.1. Define the vector $\Delta\mathbf{x} = -\mathbf{H}_m(\mathbf{x})^{-1}\nabla f(\mathbf{x})$ and the function $\tilde{f} : [0,1] \to \mathbb{R}$, $\tilde{f}(\eta) = f(\mathbf{x}+\eta\Delta\mathbf{x})$. Differentiating with respect to η, the first two derivatives of $\tilde{f}$ yield

$$\tilde{f}'(\eta) = \nabla f(\mathbf{x}+\eta\Delta\mathbf{x})^T\Delta\mathbf{x} \quad \text{and} \quad \tilde{f}''(\eta) = \Delta\mathbf{x}^T\nabla^2 f(\mathbf{x}+\eta\Delta\mathbf{x})\Delta\mathbf{x}. \tag{51}$$

Then, by virtue of the Lipschitz continuity of $\nabla^2 f(\mathbf{x})$ (cf. Assumption 1), we can upper bound the absolute value of the difference of $\tilde{f}''(\eta)$ evaluated at η and 0 by

$$|\tilde{f}''(\eta) - \tilde{f}''(0)| \leq \|\nabla^2 f(\mathbf{x}+\eta\Delta\mathbf{x}) - \nabla^2 f(\mathbf{x})\|\|\Delta\mathbf{x}\|_2^2 \leq \eta L\|\Delta\mathbf{x}\|_2^3. \tag{52}$$

The latter inequality allows us to upper bound the second derivative of $\tilde{f}(\eta)$ by

$$\tilde{f}''(\eta) \leq \tilde{f}''(0) + \eta L\|\Delta\mathbf{x}\|_2^3. \tag{53}$$

Integrating twice with respect to η on both sides of the above inequality yields

$$\tilde{f}(\eta) \leq \tilde{f}(0) + \tilde{f}'(0)\eta + \tilde{f}''(0)\frac{\eta^2}{2} + \frac{\eta^3}{6}L\|\Delta\mathbf{x}\|_2^3. \tag{54}$$

Evaluate (54) at $\eta = 1$ to get the following upper bound for $f(\mathbf{x}+\Delta\mathbf{x})$:

$$f(\mathbf{x}+\Delta\mathbf{x}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T\nabla^2 f(\mathbf{x})\Delta\mathbf{x} + \frac{L}{6}\|\Delta\mathbf{x}\|_2^3. \tag{55}$$

We next work towards an upper bound for the term $\Delta\mathbf{x}^T\nabla^2 f(\mathbf{x})\Delta\mathbf{x}$. Using the eigenvalue decomposition, it is possible to write this term as

$$\Delta\mathbf{x}^T\nabla^2 f(\mathbf{x})\Delta\mathbf{x} = \left(\mathbf{Q}(\mathbf{x})\nabla f(\mathbf{x})\right)^\top\left|\mathbf{\Lambda}(\mathbf{x})\right|_m^{-1}\mathbf{\Lambda}(\mathbf{x})\left|\mathbf{\Lambda}(\mathbf{x})\right|_m^{-1}\mathbf{Q}(\mathbf{x})\nabla f(\mathbf{x}). \tag{56}$$

Observe that by definition it holds that $\mathbf{\Lambda}(\mathbf{x}) \preceq |\mathbf{\Lambda}(\mathbf{x})|_m$; thus we have that

$$\Delta\mathbf{x}^T\nabla^2 f(\mathbf{x})\Delta\mathbf{x} \leq \left(\mathbf{Q}(\mathbf{x})\nabla f(\mathbf{x})\right)^\top |\mathbf{\Lambda}(\mathbf{x})|_m^{-1}\, \mathbf{Q}(\mathbf{x})\nabla f(\mathbf{x}) = -\nabla f(\mathbf{x})^\top \Delta\mathbf{x}. \tag{57}$$

Next, we derive an upper bound for $\|\Delta\mathbf{x}\|_2^3$. By definition of $\Delta\mathbf{x}$ we have that

$$\|\Delta\mathbf{x}\|_2^2 = \nabla f(\mathbf{x})^T\mathbf{H}_m(\mathbf{x})^{-1}\mathbf{H}_m(\mathbf{x})^{-1}\nabla f(\mathbf{x}). \tag{58}$$

Since the previous expression is maximized when $\nabla f(\mathbf{x})$ is collinear with the eigenvector corresponding to the eigenvalue with minimum absolute value, we have that

$$\|\Delta\mathbf{x}\|_2^2 \leq -\frac{1}{m}\nabla f(\mathbf{x})^T\Delta\mathbf{x}. \tag{59}$$

Combining the upper bounds derived in (55), (57), and (59) we have that

$$f(\mathbf{x}+\Delta\mathbf{x}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T\Delta\mathbf{x}\left(\frac{1}{2} - \frac{L}{6m^{3/2}}| - \nabla f(\mathbf{x})^T\Delta\mathbf{x}|^{1/2}\right). \tag{60}$$

We then write the following upper bound for $| - \nabla f(\mathbf{x})^T\Delta\mathbf{x}|^{1/2}$ using the fact that the eigenvalue with the minimum absolute value of $\mathbf{H}_m(\mathbf{x})$ is m:

$$| - \nabla f(\mathbf{x})^T\Delta\mathbf{x}|^{1/2} \leq \frac{1}{m^{1/2}}\|\nabla f(\mathbf{x})\|_2. \tag{61}$$

This allows us to upper bound (60) by

$$f(\mathbf{x}+\Delta\mathbf{x}) \leq f(\mathbf{x}) + \frac{1}{2}\nabla f(\mathbf{x})^T\Delta\mathbf{x}\left(1 - \frac{L}{3m^2}\|\nabla f(\mathbf{x})\|_2\right). \tag{62}$$

Finally, for any $\mathbf{x} \in \mathbb{R}^n$ such that $\|\nabla f(\mathbf{x})\|_2 < (3m^2/L)\,(1-2\alpha)$ we have that

$$f(\mathbf{x}+\Delta\mathbf{x}) \leq f(\mathbf{x}) + \alpha\nabla f(\mathbf{x})^T\Delta\mathbf{x}, \tag{63}$$

which shows that line search admits a step size of size 1.

Appendix C. Proof of Corollary 3.3. In the neighborhood of a local minimum all the eigenvalues of the Hessian are positive; thus (22) reduces to

$$\|\nabla f(\mathbf{x}_{k+1})\| = \|\nabla f_+(\mathbf{x}_{k+1})\| \leq \frac{5L}{2m^2}\left\|\nabla f(\mathbf{x}_k)\right\|^2. \tag{64}$$

Multiplying both sides of the previous equation by $5L/2m^2$ yields

$$\frac{5L}{2m^2}\|\nabla f(\mathbf{x}_{k+1})\| \leq \left(\frac{5L}{2m^2}\left\|\nabla f(\mathbf{x}_k)\right\|\right)^2. \tag{65}$$

The previous inequality can be written recursively as

$$\frac{5L}{2m^2}\|\nabla f(\mathbf{x}_k)\| \leq \left(\frac{5L}{2m^2}\left\|\nabla f(\mathbf{x}_0)\right\|\right)^{2^k} \leq 2^{-2^k}, \tag{66}$$

where the rightmost inequality holds since $\|\nabla f(\mathbf{x}_0)\| < \delta \leq m^2/(5L)$. Thus

$$\frac{5L}{2m^2}\|\nabla f(\mathbf{x}_{K_3})\| \leq 2^{-2^{K_3}} = \frac{5L}{2m^2}\varepsilon, \tag{67}$$

which completes the proof of the corollary.

Appendix D. Proof of Proposition 3.4. Let us start by showing that if $\|\nabla f_-(\mathbf{x}_0)\| > (5L)/(2m^2)\|\nabla f(\mathbf{x}_0)\|^2$, then $\|\nabla f_-(\mathbf{x}_1)\| > \|\nabla f_+(\mathbf{x}_1)\|$. Using the results from Theorem 3.2 we have that

$$\|\nabla f_-(\mathbf{x}_1)\| \geq 2\|\nabla f_-(\mathbf{x}_0)\| - \frac{5L}{2m^2}\|\nabla f(\mathbf{x}_0)\|^2 > \|\nabla f_-(\mathbf{x}_0)\| \geq \varepsilon \tag{68}$$

and

$$\|\nabla f_+(\mathbf{x}_1)\| \leq \frac{5L}{2m^2}\|\nabla f(\mathbf{x}_0)\|^2 < \|\nabla f_-(\mathbf{x}_0)\|. \tag{69}$$

Therefore $\|\nabla f_-(\mathbf{x}_1)\| > \|\nabla f_+(\mathbf{x}_1)\|$. We next show that the same holds for all $k \geq 1$. Let us prove it by induction. Assume $\|\nabla f_-(\mathbf{x}_k)\| \geq \|\nabla f_+(\mathbf{x}_k)\|$ holds for some $k \geq 1$. In this case we have that $\|\nabla f(\mathbf{x}_k)\| \leq 2\|\nabla f_-(\mathbf{x}_k)\|$; thus

$$\frac{5L}{2m^2}\|\nabla f(\mathbf{x}_k)\|^2 \leq \frac{5L}{m^2}\|\nabla f(\mathbf{x}_k)\|\|\nabla f_-(\mathbf{x}_k)\| \leq \zeta\|\nabla f_-(\mathbf{x}_k)\|, \tag{70}$$

where the last inequality follows from the fact that $\|\nabla f(\mathbf{x}_k)\| \leq \zeta\delta \leq \zeta(m^2)/(5L)$. Since $\|\nabla f(\mathbf{x}_k)\| \leq \zeta\delta$, the hypotheses of Theorem 3.2 are satisfied and we have that

$$\|\nabla f_-(\mathbf{x}_{k+1})\| \geq 2\|\nabla f_-(\mathbf{x}_k)\| - \frac{5L}{2m^2}\|\nabla f(\mathbf{x}_k)\|^2 \geq (2-\zeta)\|\nabla f_-(\mathbf{x}_k)\| \tag{71}$$

and

$$\|\nabla f_+(\mathbf{x}_{k+1})\| \leq \frac{5L}{2m^2}\|\nabla f(\mathbf{x}_k)\|^2 \leq \zeta\|\nabla f_-(\mathbf{x}_k)\|, \tag{72}$$

where the rightmost inequality in the two previous expressions follows from the bound (70). The latter implies that $\|\nabla f_+(\mathbf{x}_{k+1})\| \leq \|\nabla f_-(\mathbf{x}_{k+1})\|$ since $\zeta \in (0,1)$. This proves that for all $k \geq 1$ we have that $\|\nabla f_+(\mathbf{x}_k)\| \leq \|\nabla f_-(\mathbf{x}_k)\|$. The latter allows us to write (71) for every $k \geq 1$ as long as $\|\nabla f(\mathbf{x}_k)\| \leq \zeta\delta$. Therefore, writing (71) recursively, for $K_2 > 0$ it holds that

$$\|\nabla f_-(\mathbf{x}_{K_2})\| \geq (2-\zeta)^{K_2-1}\|\nabla f_-(\mathbf{x}_1)\| \geq (2-\zeta)^{K_2-1}\varepsilon, \tag{73}$$

where we used (68) to lower bound $\|\nabla f_-(\mathbf{x}_1)\| > \varepsilon$ in the rightmost inequality in the previous expression. Hence, for $\mathbf{x}_{K_1}$, with $K_2 = 1 + \log(\zeta\delta/\varepsilon)/\log(2-\zeta)$ we have

$$\|\nabla f_-(\mathbf{x}_{K_2})\| \geq (2-\zeta)^{\frac{\log(\zeta\delta/\varepsilon)}{\log(2-\zeta)}}\varepsilon = e^{\log\frac{\zeta\delta}{\varepsilon}}\varepsilon = \zeta\delta, \tag{74}$$

which completes the proof of the proposition.

Appendix E. Proof of Proposition 3.5. We consider the case in which $\|\nabla f_-(\mathbf{x}_k)\| < \frac{5L}{2m^2}\|\nabla f(\mathbf{x}_k)\|^2$ for all k. Because $\|\nabla f(\mathbf{x}_k)\| \leq \delta \leq m^2/(5L)$ we have that

$$\|\nabla f_-(\mathbf{x}_k)\| < \frac{1}{2}\|\nabla f(\mathbf{x}_k)\|. \tag{75}$$

Hence, $\|\nabla f_-(\mathbf{x}_k)\| < \|\nabla f_+(\mathbf{x}_k)\|$. This, being the case for all k, implies that

$$\|\nabla f(\mathbf{x}_{k+1})\| \leq 2\|\nabla f_+(\mathbf{x}_{k+1})\| \leq \frac{5L}{m^2}\|\nabla f(\mathbf{x}_k)\|^2, \tag{76}$$

where the last inequality follows from the result of Theorem 3.2. Then write recursively the previous expression as

$$\frac{5L}{m^2}\left\|\nabla f(\mathbf{x}_k)\right\| \leq \left(\frac{5L}{m^2}\left\|\nabla f(\mathbf{x}_k)\right\|\right)^{2^k} \leq \zeta^{2^k}. \tag{77}$$

Thus, in at most $\tilde{k} = \log_2(\log_2(\frac{5L\zeta}{m^2\varepsilon}))$ we have that $\|\nabla f(\mathbf{x}_{k+\tilde{k}})\| \leq \varepsilon$.

Appendix F. Proof of Proposition 3.6. Let $\mathbf{x}_k$ be such that $\|\nabla f(\mathbf{x}_k)\| < \varepsilon$, and let q be the probability of having $\mathbf{x}_k + X \in \mathcal{G}$, with X a Gaussian random vector where each component is given by $X_i \sim \mathcal{N}(0, 2\varepsilon/m)$. If this is the case, the rate of escape of the ε neighborhood is $2-\zeta$, and therefore we escape this region in $1 + \log(1/\gamma)/\log(2-\zeta)$ iterations. The second case is when $\mathbf{x}_k + X \notin \mathcal{G}$, which happens with probability $1-q$. Two possibilities arise in this scenario: Either $\|\nabla f(\mathbf{x}_k + X)\| < \varepsilon$ or not. If the former happens, then Algorithm 2 will perform $1+\log(1/\gamma)/\log(2-\zeta)$ iterations, and it might be the case that the argument is still in the ε neighborhood, thus requiring us to perturb the iterate again as stated by Algorithm 2. The other possibility is that either immediately after the perturbation or after the $1+\log(1/\gamma)/\log(2-\zeta)$ iterations we have $\|\nabla f(\mathbf{x}_k)\| > \varepsilon$, but $\mathbf{x}_k$ is still not in $\mathcal{G}$. In this situation, from Proposition 3.5, we have that in at most $\tilde{k}$ iterations, $\|\nabla f(\mathbf{x}_{\tilde{k}})\| < \varepsilon$, where $\tilde{k}$ is given by

$$\tilde{k} = \log_2\left(\log_2\left(\frac{5L\zeta}{m^2\varepsilon}\right)\right). \tag{78}$$

To summarize, with probability $1-q$ the perturbation X yields an iterate such that $\mathbf{x}_k + X \notin \mathcal{G}$. This being the case, after at most k^* iterations the argument is in the ε neighborhood and thus we draw a new random variable, with

$$k^* = \log_2\left(\log_2\left(\frac{5L\zeta}{m^2\varepsilon}\right)\right) + \frac{\log\left(1/\gamma\right)}{\log(2-\zeta)} + 1. \tag{79}$$

We are left to show that the number of draws needed to escape each saddle with probability $(1-p)^{1/S}$ is given by $\log(S/p)/\log(1/(1-q))$. Since the draws are independent, the probability of having $x_k \notin \mathcal{G}$ after N perturbations is given by $(1-q)^N$. Hence, we require $(1-q)^N < 1-(1-p)^{\frac{1}{S}} \approx p/S$ draws, where we have used the Taylor expansion of $(1-p)^{1/S}$ around the point $p = 0$. Taking logarithm on both sides of the previous inequality yields

$$N > \frac{\log(p/S)}{\log(1-q)} = \frac{\log(S/p)}{\log(1/(1-q))}, \tag{80}$$

thus completing the proof of the proposition.

Lemma F.1. *Let $f(\mathbf{x})$ be a function satisfying Assumptions* 1 *and* 3. *Let X be a random vector, such that $X_i \sim \mathcal{N}(0, 2\varepsilon/m)$ and $\mathbf{x}_0 \in \mathbb{R}^n$ in the neighborhood of a saddle point such that $\|\nabla f(\mathbf{x}_0)\| < \varepsilon$ with*

$$\varepsilon < \min\left\{\frac{1-\gamma}{4Ln}, \frac{\zeta\delta m}{2M\sqrt{n}+m}, \frac{2L\gamma}{5\left(2M\sqrt{n}+m\right)^2}\right\}. \tag{81}$$

Then $q = P\left(\mathbf{x}_0 + X \in \mathcal{G}\right)$ satisfies (14).

Proof. Let us define the following set for $\alpha > 1$,

$$\mathcal{G}_\alpha := \left\{ \mathbf{x} \in \mathbb{R}^n \middle| \, \|\nabla f_-(\mathbf{x})\| > \gamma\varepsilon, \|\nabla f(\mathbf{x})\| < \alpha\gamma\varepsilon, \|\nabla f_-(\mathbf{x})\| \geq \frac{5m^2}{2L} \|\nabla f(\mathbf{x})\|^2 \right\}, \tag{82}$$

and the following value for α_1,

$$\alpha_1 = \sqrt{\frac{2L}{5m^2\gamma\varepsilon}} > \frac{2M\sqrt{n}+m}{m\gamma} > \frac{1}{\gamma}. \tag{83}$$

Note as well that for every $\alpha < \alpha_1$ if $\|\nabla f_-(\mathbf{x})\| > \gamma\varepsilon$ and $\|\nabla f(\mathbf{x})\| < \alpha\gamma\varepsilon$, then $\|\nabla f_-(\mathbf{x})\| \geq \frac{5m^2}{2L} \|\nabla f(\mathbf{x})\|^2$ is satisfied. Hence, for any $\alpha < \alpha_1$ we have that

$$\mathcal{G}_\alpha := \left\{ \mathbf{x} \in \mathbb{R}^n \middle| \, \|\nabla f_-(\mathbf{x})\| > \gamma\varepsilon, \|\nabla f(\mathbf{x})\| < \alpha\gamma\varepsilon \right\}. \tag{84}$$

In addition, if $\alpha < \alpha_2 = (\zeta\delta)/(\gamma\varepsilon)$, $\mathcal{G}_\alpha \subset \mathcal{G}$. Hence for any $\alpha \leq \alpha_2$ it holds that $P(\mathbf{x}_0 + X \in \mathcal{G}) > P(\mathbf{x}_0 + X \in \mathcal{G}_\alpha)$ and, because $\varepsilon < (\zeta\delta m)/(5M\sqrt{n}+1)$,

$$\alpha_2 > \frac{2M\sqrt{n}+m}{m\gamma} > \frac{1}{\gamma}. \tag{85}$$

For what remains we choose $\alpha^* = (2M\sqrt{n}+m)/(m\gamma)$. We proceed to lower bound $\|\nabla f_-(\mathbf{x}_0 + X)\|$. From its definition (8) and the Taylor expansion around $\mathbf{x}_c$ we have

$$\nabla f_-(\mathbf{x}_0 + X) = \mathbf{Q}_-^\top \nabla f(\mathbf{x}_0 + X) = \mathbf{Q}_-^\top \nabla^2 f(\tilde{\mathbf{x}})(\mathbf{x}_0 + X - \mathbf{x}_c), \tag{86}$$

where $\tilde{\mathbf{x}} = \theta\mathbf{x} + (1-\theta)\mathbf{x}_c$ for some $\theta \in [0,1]$. Using the Lipschitz continuity of the Hessian, we can upper bound the norm of the previous expression by

$$\|\nabla f_-(\mathbf{x}_0 + X)\| \geq \left\|\mathbf{Q}_-^\top \nabla^2 f(\mathbf{x}_c)\left(\mathbf{x}_0 + X - \mathbf{x}_c\right)\right\| - L\left\|\mathbf{x}_0 + X - \mathbf{x}_c\right\|^2. \tag{87}$$

Since the absolute value of the minimum eigenvalue of the Hessian at $\mathbf{x}_c$ is at least $2m$ (cf. Assumption 3), it holds that

$$\begin{aligned} \|\nabla f_-(\mathbf{x}_0 + X)\| &\geq 2m\left\|\mathbf{Q}_-^\top\left(\mathbf{x}_0 + X - \mathbf{x}_c\right)\right\| - L\left\|\mathbf{x}_0 + X - \mathbf{x}_c\right\|^2 \\ &\geq 2m\left\|\mathbf{Q}_-^\top X\right\| - 2m\left\|\mathbf{x}_0 - \mathbf{x}_c\right\| - L\left\|\mathbf{x}_0 - \mathbf{x}_c\right\|^2 - L\left\|X\right\|^2. \end{aligned} \tag{88}$$

Recall that (cf. Appendix B) if $\|\nabla f(\mathbf{x}_0)\| < \delta$, we have that $m\left\|\mathbf{x}_0 - \mathbf{x}_c\right\| \leq \|\nabla f(\mathbf{x}_0)\| \leq m^2/L$, where the last inequality follows from the definition of δ in (7). Hence

$$2m\left\|\mathbf{x}_0 - \mathbf{x}_c\right\| + L\left\|\mathbf{x}_0 - \mathbf{x}_c\right\|^2 \leq 3m\left\|\mathbf{x}_0 - \mathbf{x}_c\right\| \leq 3\left\|\nabla f(\mathbf{x}_0)\right\| < 3\varepsilon. \tag{89}$$

Combining the previous bound with (88) yields

$$\|\nabla f_-(\mathbf{x}_0 + X)\| \geq 2m\left\|\mathbf{Q}_-^\top X\right\| - L\left\|X\right\|^2 - 3\varepsilon. \tag{90}$$

On the other hand, the Lipschitz continuity of the gradient (cf. Assumption 1) implies

$$\|\nabla f(\mathbf{x}_0 + X)\| \leq \|\nabla f(\mathbf{x}_0)\| + M\left\|X\right\| \leq \varepsilon + M\left\|X\right\|. \tag{91}$$

Then, define $\beta = (\alpha^*\gamma - 1)\varepsilon/M$. Note that $\beta > 0$ since $\alpha > 1/\gamma$ and write

$$P\left(\mathbf{x}_0 + X \in \mathcal{G}_{\alpha^*}\right) > P\left(\left\|\mathbf{Q}_-^\top X\right\| - \frac{L}{2m}\|X\|^2 > (3+\gamma)\frac{\varepsilon}{2m}, \|X\| < \beta\right). \tag{92}$$

Conditioning on $\|X\| < \beta$ we can further lower bound the previous probability by

$$P\left(\mathbf{x}_0 + X \in \mathcal{G}_{\alpha^*}\right) > P\left(\|X\| < \beta\right)\ P\left(\left\|\mathbf{Q}_-^\top X\right\| > (3+\gamma)\frac{\varepsilon}{2m} + \frac{L}{2m}\beta^2\right). \tag{93}$$

Let us next derive a lower bound for $P(\|X\| < \beta)$. Since X_i are normal with mean zero and standard deviation σ, $\|X\|^2/\sigma^2$ is Chi-squared with parameter n. Thus,

$$\begin{aligned} P\left(\|X\| < \beta\right) = P\left(\frac{\|X\|^2}{\sigma^2} < \frac{\beta^2}{\sigma^2}\right) &= P\left(\frac{\|X\|^2}{\sigma^2} < \frac{(\alpha^*\gamma - 1)^2\varepsilon^2}{M^2\sigma^2}\right) \\ &= P\left(\frac{\|X\|^2}{\sigma^2} < \frac{(\alpha^*\gamma - 1)^2 m^2}{4M^2}\right), \end{aligned} \tag{94}$$

where in the last equality we have used the fact that $\sigma := 2\varepsilon/m$. Since $\alpha^* = (2M\sqrt{n} + m)/(\gamma m)$ we have that $(\alpha^*\gamma - 1)m = 2M\sqrt{n}$. Hence

$$P\left(\|X\| < \beta\right) > P\left(\frac{\|X\|^2}{\sigma^2} < n\right) = \frac{\gamma\left(\frac{n}{2}, \frac{n}{2}\right)}{\Gamma\left(\frac{n}{2}\right)}. \tag{95}$$

Note that the ratio $\gamma(\frac{n}{2}, \frac{n}{2})/\Gamma(\frac{n}{2})$ for any $n \geq 1$ is lower bounded by $1/2$. Therefore, we obtain that $P\left(\|X\| < \beta\right) > 1/2$. Now we proceed to derive a lower bound for the second probability in the right-hand side of (99). Let l be the number of negative eigenvalues of the Hessian, and let $\mathbf{q}_i$ be the eigenvectors corresponding to those eigenvalues. Then the norm $\left\|\mathbf{Q}_-^\top X\right\|$ can be written as

$$\left\|\mathbf{Q}_-^\top X\right\| = \sqrt{\sum_{i=1}^{l} |\mathbf{q}_i^\top X|^2}. \tag{96}$$

For each i the product $\mathbf{q}_i^\top X$ is a linear combination of Gaussian random variables. Thus, $Y_i = \mathbf{q}_i^\top X \sim \mathcal{N}(\sum_{j=1}^n \mathbf{q}_{ij}\mu_j, \sum_{j=1}^n \mathbf{q}_{ij}^2\sigma_j^2)$. Since each X_i is normal with mean zero and variance σ^2 and the norm of $\mathbf{q}_i$ is one, we have that $Y_i \sim \mathcal{N}(0, \sigma^2)$. Because the minimum norm of $\mathbf{Q}_-^\top X$ is attained when there is only one negative eigenvalue, it is possible to lower bound its norm by$\left\|\mathbf{Q}_-^\top X\right\| \geq |Y_i|$. Thus we have that

$$P\left(\left\|\mathbf{Q}_-^\top X\right\| > (3+\gamma)\frac{\varepsilon}{2m} + \frac{L}{2m}\beta^2\right) = P\left(\frac{|Y_i|}{\sigma} > (3+\gamma)\frac{\varepsilon}{2m\sigma} + \frac{L}{2m\sigma}\beta^2\right). \tag{97}$$

Recall that $\sigma = 2\varepsilon/m$; thus

$$(3+\gamma)\frac{\varepsilon}{2m\sigma} + \frac{L\beta^2}{2m\sigma} = \frac{3+\gamma}{4} + \frac{L(\alpha^*\gamma - 1)^2\varepsilon}{4M^2} = \frac{3+\gamma}{4} + Ln\varepsilon < 1, \tag{98}$$

where the latter inequality comes from the fact that $\varepsilon < (1-\gamma)/(4Ln)$. Therefore,

$$P\left(\left\|\mathbf{Q}_-^\top X\right\| > (3+\gamma)\frac{\varepsilon}{2m} + \frac{L}{2m}\beta^2\right) \geq 2\left(1 - \Phi(1)\right). \tag{99}$$

Hence combining the bounds in (93), (95), and (99) we have that

$$P(\mathbf{x}_0 + X \in \mathcal{G}) > P(\mathbf{x}_0 + X \in \mathcal{G}_{\alpha^*}) > 2\left(1 - \Phi(1)\right)\frac{\gamma\left(\frac{n}{2}, \frac{n}{2}\right)}{\Gamma\left(\frac{n}{2}\right)}, \tag{100}$$

thus completing the proof of the lemma. □

Appendix G. Proof of Proposition 3.7. Use the fact that the maximum eigenvalue of the Hessian is bounded by M (cf. Assumption 1) to write

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \eta_k \nabla f(\mathbf{x}_k)^\top \Delta\mathbf{x} + \frac{\eta_k^2}{2} M \left\|\Delta\mathbf{x}\right\|_2^2. \tag{101}$$

The previous expression can be further upper bounded using (59) as

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \eta_k \nabla f(\mathbf{x}_k)^\top \Delta\mathbf{x} \left(1 - \frac{\eta_k M}{2m}\right). \tag{102}$$

Observe that $\hat{\eta} = m/M$ satisfies the exit condition of the line search. Hence, backtracking outputs a step size satisfying $\eta_k > \beta m/M$. Thus, we have that

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \alpha\eta_k \nabla f(\mathbf{x}_k)^\top \Delta\mathbf{x} \leq f(\mathbf{x}_k) + \alpha\beta\frac{m}{M}\nabla f(\mathbf{x}_k)^\top \Delta\mathbf{x}. \tag{103}$$

Observe that the maximum eigenvalue of $\nabla^2 f(\mathbf{x})$ is bounded by M; hence we have that $-\nabla f(\mathbf{x}_k)^\top \Delta\mathbf{x} \geq \left\|\nabla f(\mathbf{x}_k)\right\|^2 / M$. Combine this fact with the characterization of the region away from the saddle points, i.e., $\|\nabla f(\mathbf{x}_k)\| > \delta\zeta$, to bound (103) as

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \alpha\beta\frac{m}{M^2}(\zeta\delta)^2. \tag{104}$$

The previous inequality shows that $f(\mathbf{x})$ is decreasing along the sequence $\{\mathbf{x}_k\}$. Since $f(\mathbf{x})$ is lower bounded, the sequence $f(\mathbf{x}_k)$ converges, which by virtue of (103) implies that $\lim_{k\to\infty} \|\nabla f(\mathbf{x}_k)\| = 0$. Let $\mathbf{x}^* := \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$, and let $\mathbf{x}_k$ be the first iterate satisfying $\|\nabla f(\mathbf{x}_k)\| \leq \zeta\delta$. Then we can write

$$\begin{aligned} f(\mathbf{x}_0) - f(\mathbf{x}^*) \geq f(\mathbf{x}_0) - f(\mathbf{x}_k) &= \sum_{i=0}^{k-1} f(\mathbf{x}_i) - f(\mathbf{x}_{i+1}) \\ &\geq \sum_{i=0}^{k-1} \alpha\beta\frac{m}{M^2}(\zeta\delta)^2 = k\alpha\beta\frac{m}{M^2}(\zeta\delta)^2, \end{aligned} \tag{105}$$

where the last inequality follows from (104). Hence the number of iterations needed for Algorithm 2 to reach the neighborhood defined by $\|\nabla f(\mathbf{x}_k)\| < \zeta\delta$ is bounded by

$$k \leq \frac{M^2\left(f(\mathbf{x}_0) - f(\mathbf{x}^*)\right)}{\alpha\beta m(\zeta\delta)^2}. \tag{106}$$

Appendix H. Proof of Proposition 3.8. Let us start by showing that if we visit a saddle point multiple times, NCN adds noise to the iterates in only one of those visits. Observe that in order to perform a random draw it must hold that $\|\nabla f(\mathbf{x})\| < \varepsilon$ (cf. step 4 of Algorithm 2). Using the fact that the maximum eigenvalue of $\nabla^2 f(\mathbf{x})$ is bounded by M, we can lower bound $f(\mathbf{x})$ in that neighborhood by

$$f(\mathbf{x}) \geq f(\mathbf{x}_c) - \frac{M}{m^2}\varepsilon^2. \tag{107}$$

Hence, if we show that after adding noise and escaping the saddle, the value of the function is smaller than the right-hand side of the above equation, since the update ensures decrement, the iterates can never reach the neighborhood $\|\nabla f(\mathbf{x})\| < \varepsilon$ again.

Therefore noise is added only once. Observe that since the random noise is added to ensure that $\|\nabla f(\mathbf{x})\|$ is bounded by $(2\sqrt{n}\frac{M}{m}+1)\varepsilon$ (cf. step 6 of Algorithm 2), it is possible to upper bound $f(\mathbf{x}_k)$, with $\mathbf{x}_k$ being the iterate after a random draw, by

$$f(\mathbf{x}_k) \leq f(\mathbf{x}_c) + \frac{M}{m^2}\left(2\sqrt{n}\frac{M}{m}+1\right)^2 \varepsilon^2. \tag{108}$$

Since the line search routine ensures that the function is decreasing along the sequence of iterates generated by NCN (cf. step 4 of Algorithm 1), the value of the function of the first iterate outside the neighborhood $\mathcal{N}$ is at most $f(\mathbf{x}_c)+M/m^2\left(2\sqrt{n}\frac{M}{m}+1\right)^2\varepsilon^2$. As in the proof of Proposition 3.7 we have that the decrement in the value of $f(\mathbf{x}_k)$ in the region satisfying $\|\nabla f(\mathbf{x})\| > \zeta\delta$ is lower bounded by $\alpha\beta(m/M^2)(\zeta\delta)^2$. Hence, if we return to $\mathcal{N}$ in one iteration, it must be the case that

$$f(\mathbf{x}_k) \leq f(\mathbf{x}_c) + \frac{M}{m^2}\left(2\sqrt{n}\frac{M}{m}+1\right)^2 \varepsilon^2 - \alpha\beta\frac{m}{M^2}(\zeta\delta)^2 < f(\mathbf{x}_c) - \frac{M}{m^2}\varepsilon^2, \tag{109}$$

where the last inequality follows from the assumption on ε. Therefore the iterates of NCN never reach the neighborhood $\|\nabla f(\mathbf{x})\| < \varepsilon$ again, hence ensuring that only one draw is performed. Using the fact that the function f is decreasing along the sequence of iterates generated by NCN, it is possible to show that one can visit each neighborhood of a saddle point only a finite number of times. Let us upper and lower bound the value of $f(\mathbf{x})$ with $\mathbf{x} \in \mathcal{N}$ by

$$f(\mathbf{x}_c) - \frac{M}{m^2}(\zeta\delta)^2 \leq f(\mathbf{x}) \leq f(\mathbf{x}_c) + \frac{M}{m^2}(\zeta\delta)^2. \tag{110}$$

To make sure that the iterates escape $\mathcal{N}$ we need to ensure that the value of the function along the sequence of iterates generated by NCN decreases by $2M/m^2(\zeta\delta)^2 + M/m^2\left(2\sqrt{n}\frac{M}{m}+1\right)^2\varepsilon^2$. The worst case scenario is when after one iteration outside the region $\mathcal{N}$ we are back in it. In each one of these iterations the function value is reduced by $\alpha\beta m/M(\zeta\delta)^2$. Hence the maximum number of times that one visits the same neighborhood of a critical point is upper bounded by

$$T < \frac{M^3}{m^3}\left(\frac{2}{\alpha\beta} + \left(2\sqrt{n}\frac{M}{m}+1\right)^2 \frac{\varepsilon^2}{(\zeta\delta)^2}\right) < \frac{2}{\alpha\beta}\frac{M^3}{m^3} + \alpha\beta. \tag{111}$$

By virtue of the same argument, two consecutive visited critical points satisfy

$$f(\mathbf{x}_{c_i}) - f(\mathbf{x}_{c_{i+1}}) > \alpha\beta\frac{M}{m^2}\left(\zeta\delta\right)^2. \tag{112}$$

Thus, the maximum number of saddle points that can be visited, S, satisfies

$$f(\mathbf{x}_0) - f(\mathbf{x}^*) > S \min_{i=1\ldots S}\left(f(\mathbf{x}_{c_i}) - f(\mathbf{x}_{c_{i+1}})\right) > S\alpha\beta\frac{M}{m^2}\left(\zeta\delta\right)^2, \tag{113}$$

where $\mathbf{x}_{c_{S+1}}$ is a local minimum, thus completing the proof of the proposition.

REFERENCES

[1] N. Agarwal, Z. Allen-Zhu, B. Bullins, E. Hazan, and T. Ma, *Finding approximate local minima faster than gradient descent*, in Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, ACM, New York, 2017, pp. 1195–1199.

[2] P. Baldi and K. Hornik, *Neural networks and principal component analysis: Learning from examples without local minima*, Neural Netw., 2 (1989), pp. 53–58.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[4] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford, *Accelerated methods for nonconvex optimization*, SIAM J. Optim., 28 (2018), pp. 1751–1772, https://doi.org/10.1137/17M1114296.

[5] C. Cartis, N. I. Gould, and P. L. Toint, *Adaptive cubic regularisation methods for unconstrained optimization. Part* I: *Motivation, convergence and numerical results*, Math. Program., 127 (2011), pp. 245–295.

[6] C. Cartis, N. I. Gould, and P. L. Toint, *Adaptive cubic regularisation methods for unconstrained optimization. Part* II: *Worst-case function- and derivative-evaluation complexity*, Math. Program., 130 (2011), pp. 295–319.

[7] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, *The loss surfaces of multilayer networks*, Proceedings of AISTATS 2015, J. Mach. Learn. Res., 38 (2015), pp. 192–204.

[8] A. R. Conn, N. I. Gould, and P. L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Springer Ser. Comput. Math. 17, Springer Science & Business Media, 2013.

[9] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, SIAM, Philadelphia, 2000, https://doi.org/10.1137/1.9780898719857.

[10] F. E. Curtis and D. P. Robinson, *Exploiting Negative Curvature in Deterministic and Stochastic Optimization*, preprint, https://arxiv.org/abs/1703.00412, 2017.

[11] F. E. Curtis, D. P. Robinson, and M. Samadi, *A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization*, Math. Program., 162 (2017), pp. 1–32.

[12] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2014, pp. 2933–2941.

[13] A. S. Drud, *CONOPT—A large-scale GRG code*, ORSA J. Comput., 6 (1994), pp. 207–216.

[14] R. Ge, F. Huang, C. Jin, and Y. Yuan, *Escaping from saddle points—Online stochastic gradient for tensor decomposition*, in Proceedings of COLT, 2015, pp. 797–842.

[15] R. Ge, J. D. Lee, and T. Ma, *Matrix completion has no spurious local minimum*, in Advances in Neural Information Processing Systems 29, Curran Associates, 2016, pp. 2973–2981.

[16] N. I. Gould and J. Nocedal, *The modified absolute-value factorization norm for trust-region minimization*, in High Performance Algorithms and Software in Nonlinear Optimization, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998, pp. 225–241.

[17] J. Greenstadt, *On the relative efficiencies of gradient methods*, Math. Comp., 21 (1967), pp. 360–367.

[18] A. Griewank, *The Modification of Newton's Method for Unconstrained Optimization by Bounding Cubic Terms*, Technical Report NA/12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK, 1981.

[19] F. M. Harper and J. A. Konstan, *The MovieLens datasets: History and context*, ACM Trans. Interact. Intell. Syst., 5 (2016), 19.

[20] P. Jain and P. Kar, *Non-convex optimization for machine learning*, Found. Trends Mach. Learn., 10 (2017), pp. 142–336, https://doi.org/10.1561/2200000058.

[21] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, *How to escape saddle points efficiently*, in Proceedings of the 34th International Conference on Machine Learning, PMLR 70, 2017, pp. 1724–1732.

[22] K. Kawaguchi, *Deep learning without poor local minima*, in Advances in Neural Information Processing Systems 29, Curran Associates, 2016, pp. 586–594.

[23] M. Kočvara and M. Stingl, *Pennon: A code for convex nonlinear and semidefinite programming*, Optim. Methods Softw., 18 (2003), pp. 317–333.

[24] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, *Gradient descent only converges to minimizers*, in Proceedings of the 29th Annual Conference on Learning Theory, 2016, pp. 1246–1257.

[25] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572, https://doi.org/10.1137/0904038.

[26] W. Murray, *Newton-type methods*, in Wiley Encyclopedia of Operations Research and Management Science, John Wiley & Sons, 2010; available online at https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470400531.eorms0569.

[27] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Appl. Optim. 87, Springer Science & Business Media, 2004.

[28] Y. Nesterov and B. T. Polyak, *Cubic regularization of Newton method and its global performance*, Math. Program., 108 (2006), pp. 177–205.

[29] J. Nocedal and S. Wright, *Numerical Optimization,* Springer Science & Business Media, New York, 2006.

[30] I. Panageas and G. Piliouras, *Gradient Descent Only Converges to Minimizers: Non-Isolated Critical Points and Invariant Regions*, preprint, https://arxiv.org/abs/1605.00405, 2016.

[31] R. Pascanu, Y. N. Dauphin, S. Ganguli, and Y. Bengio, *On the Saddle Point Problem for Non-Convex Optimization*, preprint, https://arxiv.org/abs/1405.4604, 2014.

[32] R. Pemantle, *Nonconvergence to unstable points in urn models and stochastic approximations*, Ann. Probab., 18 (1990), pp. 698–712.

[33] M. Powell, *On the global convergence of trust region algorithms for unconstrained minimization*, Math. Programming, 29 (1984), pp. 297–303.

[34] A. M. Saxe, J. L. McClelland, and S. Ganguli, *Exact Solutions to the Nonlinear Dynamics of Learning in Deep Linear Neural Networks*, preprint, https://arxiv.org/abs/1312.6120, 2013.

[35] A. M. Saxe, J. L. McClelland, and S. Ganguli, *Learning hierarchical category structure in deep neural networks*, in Proceedings of the 35th Annual Meeting of the Cognitive Science Society, 2013, pp. 1271–1276.

[36] J. Sun, Q. Qu, and J. Wright, *Complete dictionary recovery over the sphere* I: *Overview and the geometric picture*, IEEE Trans. Inform. Theory, 63 (2017), pp. 853–884.

[37] R. J. Vanderbei, *LOQO: An interior point code for quadratic programming. Interior point methods*, Optim. Methods Softw., 11/12 (1999), pp. 451–484.