Reading Report

— How to Escape Saddle Points Efficiently

Summary:

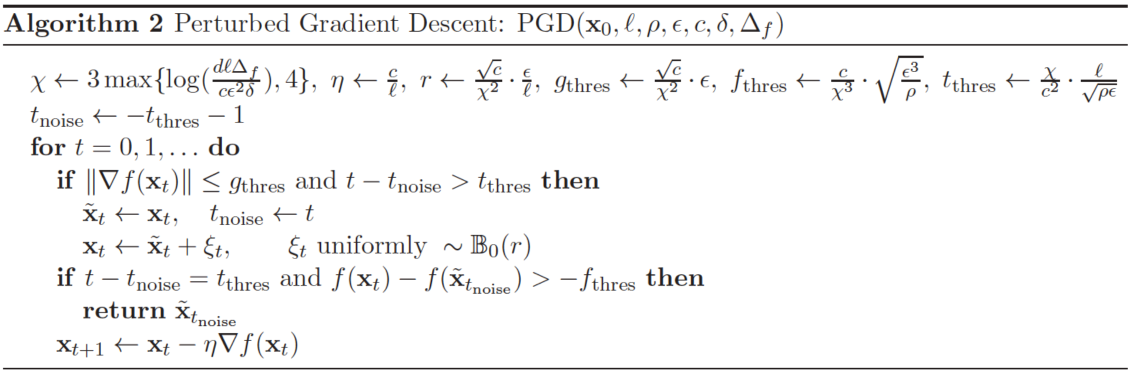
This paper proposes a perturbed version of the gradient descent method in order to escape saddle points under the assumption of strict-saddle condition. The $\textbf{Perturbed Gradient Descent(PGD)}$ detects whether the current iterate is potentially near a saddle point by the norm of the current gradient. If the norm is small enough, $\textbf{PGD}$ will add a random perturbation to the gradient, and if this operation does not make the function value decrease enough, this algorithm will be terminated. It's almost "dimension-free" (only poly-logarithmically on dimension). The main idea of this paper is that all second-order stationary points are exactly local minima. It can be proved that the convergence result is $\tilde{O}(\frac{l(f(x_0)-f^*)}{\hat{\epsilon}^2})$. This convergency rate improves previous results based on the gradient and matches the guarantee for converging to first-order stationary points up to polylog factors, which shows that this algorithm can escape saddle points almost for free. Moreover, it shows the convergence rate can be linear under the assumption of local strongly convexity and smoothness. All the above above results rely on a new characterization of the geometry around saddle points: points from where the gradient descent gets stuck at a saddle point constitute a thin "band". Using the random perturbation, iterations are very likely to escape the band. In other words, it is possible to escape from the saddle point.

Main Question: Can gradient descent escape saddle points and converge to local minima in a number of iterations that is (almost) dimension-free?

Main idea: When all saddle points are strict, all second-order stationary points are exactly local minima. —> Find a second-order stationary point.

Main result:

- **Assumption: l-smooth and ρ-Hessian Lipschitz**



Algorithm 2 Perturbed Gradient Descent: $\mathrm{PGD}(\mathbf{x}_0, \ell, \rho, \epsilon, c, \delta, \Delta_f)$

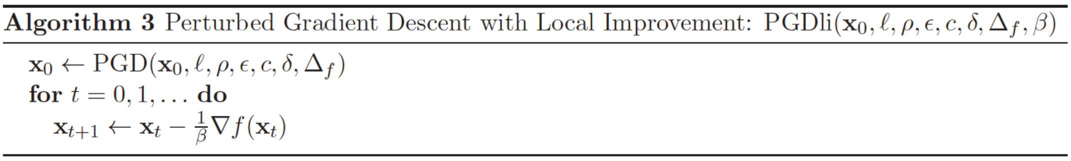
```
$\chi \leftarrow 3\max\{\log(\frac{d\ell\Delta_f}{c\epsilon^2\delta}), 4\},\ \eta \leftarrow \frac{c}{\ell},\ r \leftarrow \frac{\sqrt{c}}{\chi^2}\cdot\frac{\epsilon}{\ell},\ g_{\text{thres}} \leftarrow \frac{\sqrt{c}}{\chi^2}\cdot\epsilon,\ f_{\text{thres}} \leftarrow \frac{c}{\chi^3}\cdot\sqrt{\frac{\epsilon^3}{\rho}},\ t_{\text{thres}} \leftarrow \frac{\chi}{c^2}\cdot\frac{\ell}{\sqrt{\rho\epsilon}}$
$t_{\text{noise}} \leftarrow -t_{\text{thres}} - 1$
for $t = 0, 1, \ldots$ do
    if $\|\nabla f(\mathbf{x}_t)\| \le g_{\text{thres}}$ and $t - t_{\text{noise}} > t_{\text{thres}}$ then
        $\tilde{\mathbf{x}}_t \leftarrow \mathbf{x}_t, \quad t_{\text{noise}} \leftarrow t$
        $\mathbf{x}_t \leftarrow \tilde{\mathbf{x}}_t + \xi_t, \qquad \xi_t$ uniformly $\sim \mathbb{B}_0(r)$
    if $t - t_{\text{noise}} = t_{\text{thres}}$ and $f(\mathbf{x}_t) - f(\tilde{\mathbf{x}}_{t_{\text{noise}}}) > -f_{\text{thres}}$ then
        return $\tilde{\mathbf{x}}_{t_{\text{noise}}}$
    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta\nabla f(\mathbf{x}_t)$
```

  - This algorithm is based on gradient descent with step size η .
    - When norm of the current gradient is small, which indicates that the current iterate $\hat{x}_t$ is potentially near a saddle point, then add a small random

        perturbation to the gradient
        - The perturbation is added at most only once every t_{thres} iterations.
        - The function value does not decrease enough after t_{thres} iterations, output $\hat{x}_{t_{noise}}$
    - Convergency :
        - let $\delta > 0, \Delta_f \geq f(x_0) - f^*, c \leq c_{max}$ output ϵ-second-order stationary point, with probability $1 - \delta$, and terminate in the following number of iterations: $\tilde{O}(\frac{l(f(x_0)-f^*)}{\epsilon^2})$

- **+ strict saddle property:**
    - **Assumption: (θ, γ, ξ)-strict saddle. For any x, at least one of following holds:**
        - $\lambda_{min}(\nabla^2 f(x)) \leq -\gamma$
        - **x is ξ-close to X^*-the set of local minima**
            - $\|\nabla f(x)\| \geq \theta$
    - Convergency:
        - let $\delta > 0, \Delta_f \geq f(x_0) - f^*, c \leq c_{max}$, $\hat{\epsilon} = \min(\theta, \gamma^2/\rho)$, output a point ξ-close to X^* , with probability $1 - \delta$, and terminate in the following number of iterations: $\tilde{O}(\frac{l(f(x_0)-f^*)}{\hat{\epsilon}^2})$

- **+local structure (assumption a or b):**
    - **Assumption a. In a ξ-neighborhood of the set of local minima X^* , the function f is α-strongly convex and β-smooth.**
    - **Assumption b. In a ξ-neighborhood of the set of local minima X^* , the function f satiesfies a (α, β)-regularity condition if for any x in this neighbourhood:** $\langle \nabla f(x), x - P_{X^*}(x) \rangle \geq \frac{\alpha}{2}\|x - P_{X^*}(x)\|^2 + \frac{1}{2\beta}\|\nabla f(x)\|^2$



Algorithm 3 Perturbed Gradient Descent with Local Improvement: PGDli$(\mathbf{x}_0, \ell, \rho, \epsilon, c, \delta, \Delta_f, \beta)$

```
$\mathbf{x}_0 \leftarrow \mathrm{PGD}(\mathbf{x}_0, \ell, \rho, \epsilon, c, \delta, \Delta_f)$
for $t = 0, 1, \ldots$ do
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \frac{1}{\beta}\nabla f(\mathbf{x}_t)$
```

    - Convergency:
        - let $\delta > 0, \Delta_f \geq f(x_0) - f^*, c \leq c_{max}$, $\hat{\epsilon} = \min(\theta, \gamma^2/\rho)$, output a point ξ-close to X^* , with probability $1 - \delta$, and terminate in the following number of iterations: $O(\frac{\alpha}{\beta}\log(\frac{\xi}{\epsilon}))$