A Generic Approach for Escaping Saddle Points

reading group
present by Hongwei Jin

March 2, 2018

Problem

Nonconvex finite-sum problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \quad f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x})$$

- neither f nor f_i are necessarily convex
- assumptions
  - Lipschitz continuity of gradient on each function

$$\|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$$

  - Lipschitz continuity of Hessian

$$\left\|\nabla^2 f(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{y})\right\| \leq M\|\boldsymbol{x} - \boldsymbol{y}\|$$

Definitions

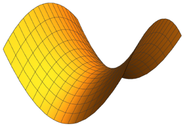
- 1st-order stationary point

$$\|\nabla_f(\boldsymbol{x})\| \leq \varepsilon$$

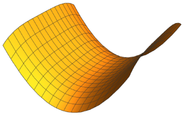
$\boldsymbol{x}$ can be a local minimum, local maximum, or a saddle point

- strict-saddle point

$$\|\nabla_f(\boldsymbol{x})\| \leq \varepsilon \quad \& \quad \lambda_{\min} \nabla^2 f(\boldsymbol{x}) < 0$$



Strict saddle point



Non-strict saddle point

- 2nd-order stationary point

$$\|\nabla_f(\boldsymbol{x})\| \leq \varepsilon \quad \& \quad \lambda_{\min} \nabla^2 f(\boldsymbol{x}) > -\gamma$$

Definitions

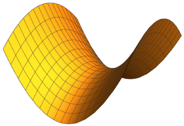
- 1st-order stationary point

$$\|\nabla_f(\boldsymbol{x})\| \leq \varepsilon$$

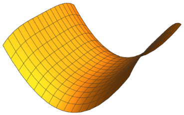
$\boldsymbol{x}$ can be a local minimum, local maximum, or a saddle point

- strict-saddle point

$$\|\nabla_f(\boldsymbol{x})\| \leq \varepsilon \quad \& \quad \lambda_{\min}\nabla^2 f(\boldsymbol{x}) < 0$$



Strict saddle point



Non-strict saddle point

- 2nd-order stationary point

$$\|\nabla_f(\boldsymbol{x})\| \leq \varepsilon \quad \& \quad \lambda_{\min}\nabla^2 f(\boldsymbol{x}) > -\gamma$$

How to get a fairly GOOD solution?

Background

Provable global optimum

- Low-rank matrix problems (algorithm independent)
  - matrix completion [Ge-Lee-Ma, NIPS'16]
    all local minima are global minima in the symmetric matrix completion problem
  - matrix sensing, matrix completion and robust PCA [Ge-Jin-Zheng, ICML'17]
    1) all local optima are global optima 2) no high-order saddle points
- Neural network
  - deep learning without poor local minima [Kawaguchi, NIPS'16]
    square loss with any depth any width: 1) local minima are global minima 2) if critical point is not global, then it's a saddle 3) exist 'bad' saddle (Hessian has no negative eigenvalue) for deeper network (more than 3 layers)
  - two-layer NN with ReLU [Li-Yuan, NIPS'17]
    input follows Gaussian dist. with standard $O(1/\sqrt{d})$ init. of weights, SGD converges to global optima
  - global optimality conditions for DNN [Yun-Sra-Jabdabaie, accepted ICLR'18]
    provide necessary and sufficient conditions for global optimality

Background Cont.

ε-approximate local minimum

- Escape strict saddle using gradient
  - SGD can escape saddle [Ge-Huang-Jin-Yuan, COLT'15]
    Noise SGD can escape saddle in the orthogonal tensor decomposition problem
  - Gradient descent converges to minimizers [Lee-Simchowitz-Jordan-Recht, COLT'16]
    GD converge to minimizer or negative infinity, proved by stable manifold theorem
  - PGD can escape saddle [Jin-Ge-Netrapalli-Kakade-Jordan, ICML'17]
    Add perturbation when enter the stuck region
- Escape saddle using Hessian explicitly
  - Cubic regularization [Nesterov-Polyak, MP'06]
- Escape strict saddle using gradient and Hessian information
  - AGD and proximal eigenvector of Hessian [Carmon-Duchi-Hinder-Sidford, arXiv'17]
    Run PCA to estimate the smallest eigenvector of Hessian and apply AGD to decrease
  - AllenZhu's works: FastCubic, Natasha2, Katyusha X, Neon [AllenZhu, arXiv'17-18]
  - Alternate between gradient and Hessian descent [Reddi-Zaheer-Sra-Poczos-Bash-Salakhutdinov-Smola, arXiv'17]
    Provide a general framework combining gradient and Hessian, and apply SVRG + HD/CR to prove the complexities

Second order Stationary Point

Definition

- An Incremental First-order Oracle (IFO) takes an index $i \in [x]$ and a point $x \in \mathbb{R}^d$, and returns the pair $(f_i(x), \nabla f_i(x))$.
- An Incremental Second-order Oracle (ISO) takes an index $i \in [x]$, a point $x \in \mathbb{R}^d$ and vector $v \in \mathbb{R}^d$, and returns the vector $\nabla^2 f_i(x)v$.

Pearlmutter's algorithm

$$\nabla f(\boldsymbol{x} + r\boldsymbol{v}) \approx \nabla f(\boldsymbol{x}) + r\nabla^2 f(\boldsymbol{x})\boldsymbol{v}$$

$$\nabla^2 f(\boldsymbol{x})\boldsymbol{v} \approx \frac{\nabla f(\boldsymbol{x} + r\boldsymbol{v}) - \nabla f(\boldsymbol{x})}{r}$$

$$\text{in practice} \quad \nabla^2 f(\boldsymbol{x})\boldsymbol{v} \approx \frac{\nabla f(\boldsymbol{x} + r\boldsymbol{v}) - \nabla f(\boldsymbol{x} - r\boldsymbol{v})}{2r}$$

Generic Framework

Idea

Interleave two subroutines to obtain a second-order critical point

- `Gradient-Focused-Optimizer`
  use the gradient information to decrease the function value
- `Hessian-Focused-Optimizer`
  use the Hessian information to avoid saddle point

Generic Framework (cont.)

Algorithm 1 Generic Framework

Input - Initial point: x^0, total iterations T, error threshold parameters ϵ, γ and probability p
for $t = 1$ **to** T **do**
    $(y^t, z^t) = \text{GRADIENT-FOCUSED-OPTIMIZER}(x^{t-1}, \epsilon)$ (refer to **G.1** and **G.2**)
    Choose u^t as y^t with probability p and z^t with probability $1 - p$
    $(x^{t+1}, \tau^{t+1}) = \text{HESSIAN-FOCUSED-OPTIMIZER}(u^t, \epsilon, \gamma)$ (refer to **H.1** and **H.2**)
    if $\tau^{t+1} = \varnothing$ **then**
        Output set $\{x^{t+1}\}$
    end if
end for
Output set $\{y^1, ..., y^T\}$

- G.1: $\mathrm{E}[f(\boldsymbol{y})] \leq f(\boldsymbol{x})$
- G.2: $\mathrm{E}\left[\|\nabla_f(\boldsymbol{y})\|^2\right] \leq \frac{1}{g(n,\epsilon)}\mathrm{E}[f(\boldsymbol{x}) - f(\boldsymbol{z})]$,
  where g is positive function: $\mathbb{N} \times \mathbb{R}^+ \to \mathbb{R}^+$
- H.1: $\mathrm{E}[f(\boldsymbol{y})] \leq f(\boldsymbol{x})$
- H.2: $\mathrm{E}[f(\boldsymbol{y})] \leq f(\boldsymbol{x}) - h(n, \epsilon, \gamma)$ when $\lambda_{\min}(\nabla^2 f(\boldsymbol{x})) \leq -\gamma$ for some h.

Main Theorem

Theorem

Let $\Delta = f(x^0) - B$ *and* $\theta = \min\left((1-p)\epsilon^2 g(n,\epsilon), ph(n,\epsilon,\gamma)\right)$. *Also, let set* Γ *be the output of Algorithm with* `Gradient-Focused-Optimizer` *satisfying G.1 and G.2 and* `Hessian-Focused-Optimizer` *satisfying H.1 and H.2. Furthermore,* T *be such that* $T > \Delta/\theta$. *Suppose the multiset* $S = \{i_1, ..., i_k\}$ *are* k *indices selected independently and uniformly randomly from* $\{1, ..., |\Gamma|\}$. *Then the following holds for the indices in* S*:*

- y^t, *where* $t \in \{i_1, ..., i_k\}$ *is a* (ϵ, γ)*-critical point with probability at least* $1 - \Delta/(T\theta)$.
- *If* $k = O\left(\frac{\log(1/\zeta)}{\log(\Delta/(T\theta))}\right)$, *with at least probability* $1 - \zeta$, *at least one iterate* y^t *where* $t \in \{i_1, ..., i_k\}$ *is a* (ϵ, γ)*-critical point.*

Gradient-Focused-Optimizer: SVRG

Algorithm 2 SVRG(x^0, ϵ)

Input: $x_m^0 = x^0 \in \mathbb{R}^d$, epoch length m, step sizes $\{\eta_i > 0\}_{i=0}^{m-1}$, iterations T_g, $S = \lceil T_g/m \rceil$
for $s = 0$ **to** $S - 1$ **do**
    $\tilde{x}^s = x_0^{s+1} = x_m^s$
    $g^{s+1} = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\tilde{x}^s)$
    for $t = 0$ **to** $m - 1$ **do**
        Uniformly randomly pick i_t from $\{1, \ldots, n\}$
        $v_t^{s+1} = \nabla f_{i_t}(x_t^{s+1}) - \nabla f_{i_t}(\tilde{x}^s) + g^{s+1}$
        $x_{t+1}^{s+1} = x_t^{s+1} - \eta_t v_t^{s+1}$
    end for
end for
Output: (y, z) where y is Iterate x_a chosen uniformly random from $\{\{x_t^{s+1}\}_{t=0}^{m-1}\}_{s=0}^{S-1}$ and $z = x_m^S$.

Lemma

Suppose $\eta_t = \eta = 1/4Ln^{2/3}$, $m = n$ and $T_g = T_\epsilon$, which depends on ϵ, then SVRG is a `Gradient-Focused-Optimizer` *with $g(n, \epsilon) = T_\epsilon / 40Ln^{2/3}$*

Hessian-Focused-Optimizer: HessianDescent

Algorithm 3 HESSIANDESCENT (x, ϵ, γ)

1: Find v such that $\|v\| = 1$, and with probability at least ρ the following inequality holds: $\langle v, \nabla^2 f(x) v\rangle \leq \lambda_{min}(\nabla^2 f(x)) + \frac{\gamma}{2}$.
2: Set $\alpha = |\langle v, \nabla^2 f(x) v\rangle|/M$.
3: $u = x - \alpha \, \text{sign}(\langle v, \nabla f(x)\rangle) v$.
4: $y = \arg\min_{z \in \{u,x\}} f(z)$
5: **Output:** $(y, \diamond)$.

Lemma

HessianDescent is a Hessian-Focused-Optimizer with $h(n, \epsilon, \gamma) = \frac{\rho}{24M^2}\gamma^3$.

Proposition

The time complexity of finding $v \in \mathbb{R}^d$ *that* $\|v\| = 1$*, and with probability at least* ρ *the following inequality holds:* $\langle v, \nabla^2 f(x) v \rangle \leq \lambda_{\min}(\nabla^2 f(x)) + \frac{\gamma}{2}$ *is* $O(nd + n^{3/4} d/\gamma^{1/2})$.

Hessian-Focused-Optimizer: HessianDescent (cont.)

Theorem

Suppose SVRG with $m = n, \eta_t = \eta = 1/4Ln^{2/3}$ for all $t \in \{1, ..., m\}$ and $T_g = \frac{40Ln^{2/3}}{\epsilon^{1/2}}$ is used as `Gradient-Focused-Optimizer` and `HessianDescent` is used as `Hessian-Focused-Optimizer` with $q = 0$, then Algorithm finds a $(\epsilon, \sqrt{\epsilon})$-second order critical point in $T = O(\frac{\Delta}{\min(p, 1-p)\epsilon^{3/2}})$ with probability at least 0.9.

Corollary

The overall running time of algorithm to find a $(\epsilon, \sqrt{\epsilon})$-second order critical point with parameter settings used in Theorem 2, is $O(nd/\epsilon^{3/2} + n^{3/4}d/\epsilon^{7/4} + n^{2/3}d/\epsilon^2)$

Hessian-Focused-Optimizer: CubicDescent

Cubic Regularization

$$\boldsymbol{v} = \arg\min_{\boldsymbol{v}} \langle \nabla f(\boldsymbol{x}), \boldsymbol{v} \rangle + \frac{1}{2} \left\langle \boldsymbol{v}, \nabla^2 f(\boldsymbol{v}) \boldsymbol{v} \right\rangle + \frac{M}{6} \|\boldsymbol{v}\|^3, \quad \boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{v}$$

Theorem

Suppose SVRG with $m = n, \eta_t = \eta = 1/4Ln^{2/3}$ for all $t \in \{1, ..., m\}$ and $T_g = \frac{40Ln^{2/3}}{\epsilon^{1/2}}$ is used as `Gradient-Focused-Optimizer` and `CubicDescent` is used as `Hessian-Focused-Optimizer` with $q = 0$, then Algorithm finds a $(\epsilon, \sqrt{\epsilon})$-second order critical point in $T = O(\frac{\Delta}{\min(p, 1-p)\epsilon^{3/2}})$ with probability at least 0.9.

Corollary

The overall running time of algorithm to find a $(\epsilon, \sqrt{\epsilon})$-second order critical point with parameter settings used in Theorem 3, is $O(nd^w/\epsilon^{3/2} + n^{2/3}d/\epsilon^2)$

Overall

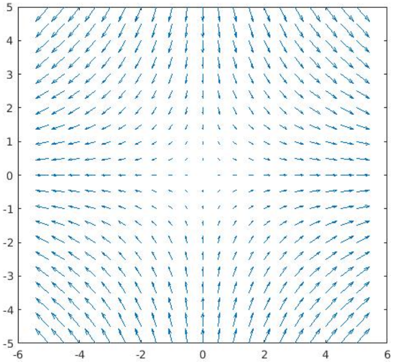
	GFO	HFO		Overall
		Iteration	Comp. per iter.	
SVRG + HD	$O(\frac{nd}{\epsilon^{3/2}} + \frac{n^{3/4}d}{\epsilon^2})$	$O(\frac{1}{\epsilon^{3/2}})$	$O(nd + \frac{n^{3/4}d}{\epsilon^{1/4}})$	$O(\frac{nd}{\epsilon^{3/2}} + \frac{n^{3/4}d}{\epsilon^{7/4}} + \frac{n^{2/3}d}{\epsilon^2})$
SVRG + CD	$O(\frac{nd}{\epsilon^{3/2}} + \frac{n^{3/4}d}{\epsilon^2})$	$O(\frac{1}{\epsilon^{3/2}})$	$O(nd^w)$	$O(\frac{nd^w}{\epsilon^{3/2}} + \frac{n^{2/3}d}{\epsilon^2})$

Algorithms[1]

point	Algorithm	Complexity (non-convex)	Hessian info.
Approx. sta. pt.	GD	$O(\frac{nd}{\epsilon^2})$	NO
Approx. sta. pt.	SGD	$O(\frac{d}{\epsilon^4})$	NO
Approx. sta. pt.	SVRG	$O(nd + \frac{n^{2/3}d}{\epsilon^2})$	NO
Approx. local min.	perturbed SGD	$O(\frac{d^C}{\epsilon^4})$	NO
Approx. local min.	cubic regularization	$O(\frac{nd^{w-1}+nd^w}{\epsilon^{3/2}})$	Yes (explicit)
Approx. local min.	FastCubic	$O(\frac{nd}{\epsilon^{3/2}} + \frac{n^{3/4}d}{\epsilon^{7/4}})$	Yes
Approx. local min.	AGD+NCD	$O(\frac{nd}{\epsilon^{3/2}} + \frac{n^{3/4}d}{\epsilon^{7/4}})$	Yes
Approx. local min.	SVRG + HD	$O(\frac{nd}{\epsilon^{3/2}} + \frac{n^{3/4}d}{\epsilon^{7/4}} + \frac{n^{2/3}d}{\epsilon^2})$	Yes
Approx. local min.	SVRG + CD	$O(\frac{nd^w}{\epsilon^{3/2}} + \frac{n^{2/3}d}{\epsilon^2})$	Yes

[1] May subject to change

Open Questions



- GFO: SVRG, Adam, SMD, etc.
  How to analysis the performance?
- HFO: acceleration of cubic?
- only first-order oracle? without Hessian-vector product?
- how to handle the “flat” saddle problem?