SGD at saddle points and shallow local minima

Speaker: Yi-Shan Wu

Institute of Information Science

Academia Sinica

Taiwan

July 06, 2018

Works

1. Escaping From Saddle Points – Online Stochastic Gradient for Tensor Decomposition 2015 [Ge, 15]
2. Efficient approaches for escaping higher order saddle points in non-convex optimization, 2016 [Ge, 16]
3. Gradient Descent Can Take exponential time to escape saddle points [Jin, 17v2]
4. How to Escape Saddle Points Efficiently, 2017 ICML [Jin, 17]
5. Escaping Saddles with Stochastic Gradients, 2018 ICML [Hadi, 18]

Questions

- Why is saddle problems so attractive ?
- Why is (varient) SGD algorithm so powerful ?
- What results we have now ?
- ...

GD and SGD

$\mathbf{w}$: parameters for neural networks
n : number of training samples
Goal : $min_{\mathbf{w}} F(\mathbf{w})$

- Gradient Descent

$$F(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{w}, z_i)$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla F(\mathbf{w}_t)$$

- Stochastic Gradient Descent

$$z_i \sim \mathcal{S}, \quad \mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla f(\mathbf{w}_t, z_i)$$

GD and SGD on Convex functions

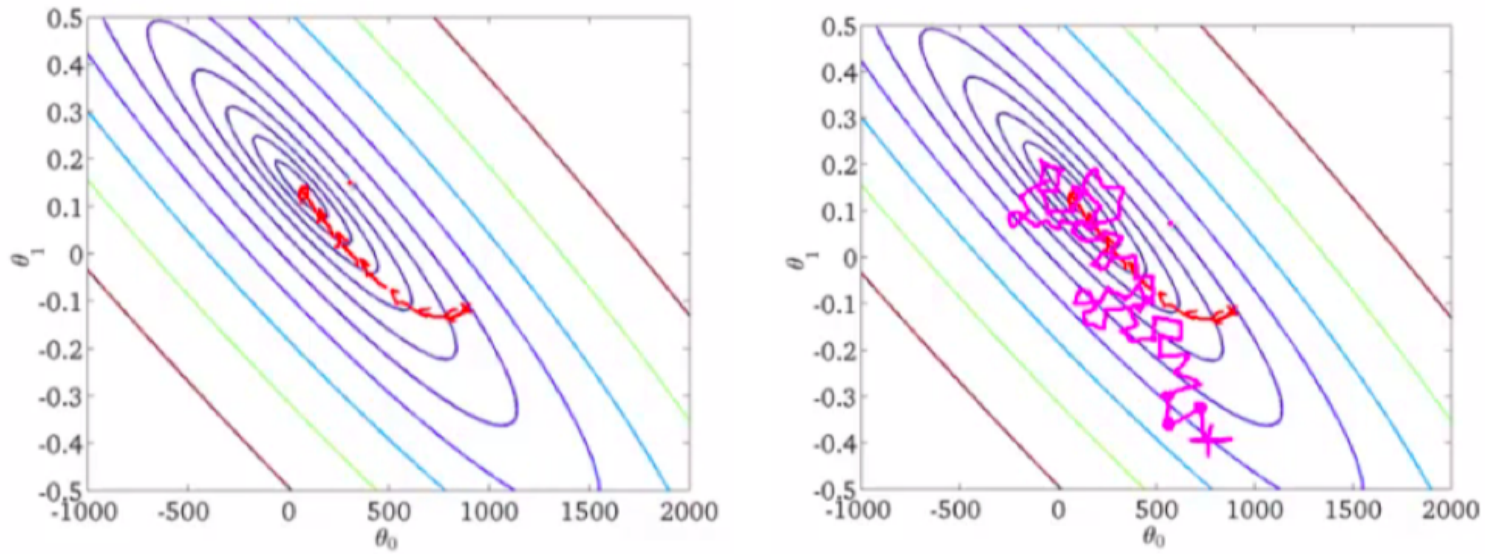


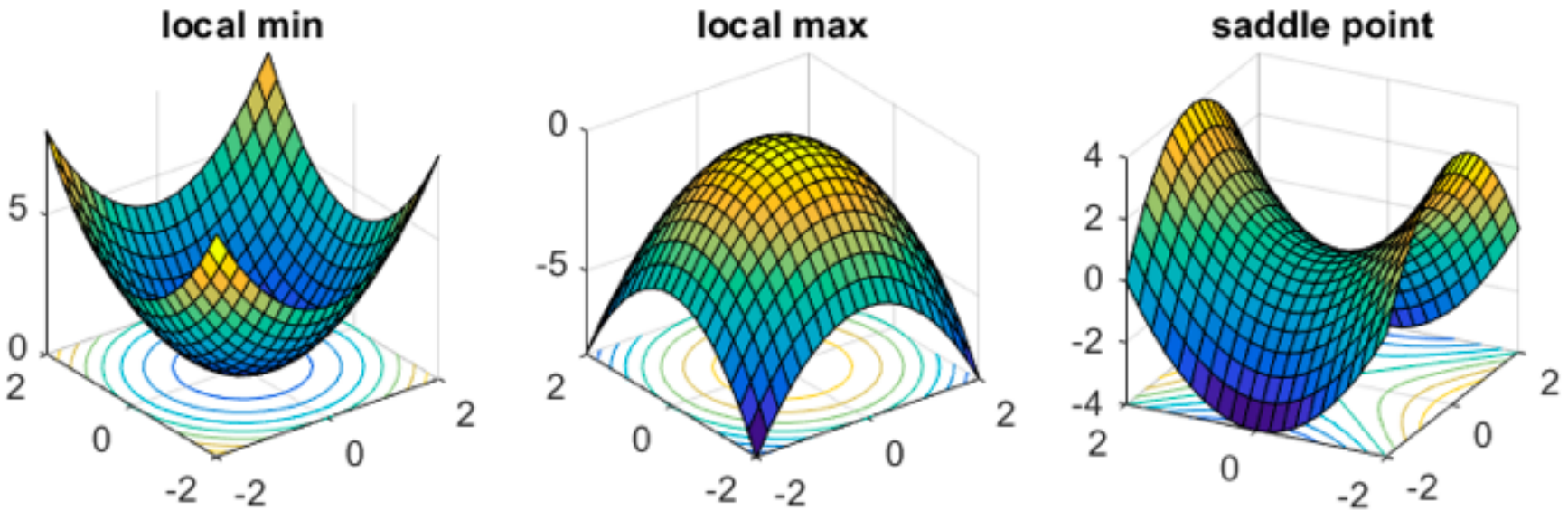
FIGURE – GD v.s. SGD

Both of them converge to local optimum (global optimum) [Shalev-Shwartz. 2009] but it is much faster to run SGD than GD.

Non-Convex functions

Optimizing a non-convex function is NP-hard in general

1. It's NP hard for even a degree 4 polynomial [Hillar & Lim, 2013]
2. Even finding a local minimum might be hard as there can be many saddle points.



Non-Convex functions

Fortunately, for many non-convex problems, it is sufficient to find a local minimum.

- All local minima are global minima : tensor decomposition, phase retrieval, matrix sensing, matrix completion, $\cdots$
- The local minima are almost as good as global minima : [Choromanska et al., 2014].

However, the bottleneck for solving non-convex functions is the great amount of "saddle points". (There can be exponentially many saddle points !)

Saddle points

Problems that have exponentially many saddle points :
tensor decomposition, dictionary learning, phase retrieval, matrix sensing, matrix completion, $\cdots$

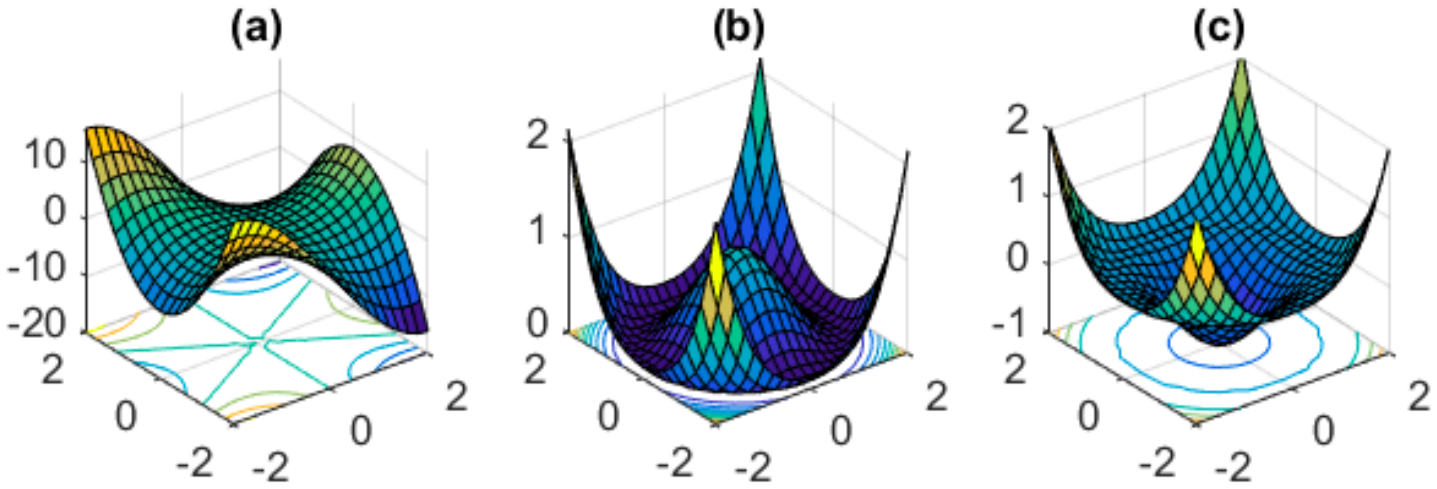


FIGURE – Monkey saddle (a), connected saddle points (b)(c)

Symmetry and Saddle points

Many learning problems can be abstracted as searching for k distinct components. Ex. clustering problems.
That is, to find k centers to minimize the distance to n points.
Obj function :

$$min_{\mathbf{x}} f(x_1, x_2, \cdots, x_k)$$

It is obvious that the solutions are permutation symmetric. That is,

$$f(x_1, x_2, \cdots, x_k) = f(x_2, x_1, \cdots, x_k)$$

However, the average of these two points is not optimal ! !

$$f(\frac{x_1 + x_2}{2}, \frac{x_1 + x_2}{2}, \cdots, x_k)$$

Short Summary

1. SGD run faster than GD.
2. Solving non-convex is hard in general since sub-optimal solutions and saddle points.
3. GD get stuck at saddle points.
4. Some problems have exponentially many saddle points but all local minima are global minima.

That means, if we can find a way to escape saddle points, we get to local minima $\Rightarrow$ global minima.
$\Rightarrow$ We are going to show that SGD can make it and converge to local minima in certain iterations.

Hessian

To escape from saddle points, we need higher order derivative since we have $\nabla F(\mathbf{w}) = 0$ at stationary points.

1. If $\nabla^2 F(\mathbf{w}) \succ 0$, it's a local minimum.
2. If $\nabla^2 F(\mathbf{w}) \prec 0$, it's a local maximum.
3. If $\nabla^2 F(\mathbf{w})$ has both $+/-$ eigenvalues, it's a saddle point.
4. (Degenerate) $\nabla^2 F(\mathbf{w})$ has eigenvalues equal to 0, it can be a local optimum or saddle point.

Idea : use 2nd order information

Taylor's expansion :

$$f(y) \approx f(x) + < \nabla f(x), y - x > + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x).$$

- If we are not at local minimum, there will be a negative eigenvalue. Therefore, even when $\nabla f(x) = 0$, we can still make progress.
- Trust region algorithms[Dauphin et al., 2014] use this idea to escape in $O(1/\epsilon^{1.5})$ iterations for "nice functions (strict saddle)".

However, costly in memory and time per iteration for the need of full Hessian matrix.

Gradient-based algorithm

We need only first-order derivative in gradient-based algorithm.
However, we know that GD may be stuck at stationary points, we need some noise. (either implicit or explicit)
Results :

Algorithm	Iterations	method
Ge, 15	$O(poly(d/\epsilon))$	noisy SGD
Levy, 16	$O(d^3 \cdot poly(1/\epsilon))$	normalized GD
Lee, 16	no guarantee	random initialization
Jin, 17	$O(polylog(d)/\epsilon^2)$	perturbed GD
Hadi, 18	d-free	pure SGD

Some Definitions

- $f : \mathbb{R}^d \to \mathbb{R}$
  - L-Lipschitz, i.e.,

$$|f(w_1) - f(w_2)| \leq L\|w_1 - w_2\|_2$$

  - ℓ-Smoothness: The gradient is ℓ-Lipschitz, i.e.

$$\|\nabla f(w_1) - \nabla f(w_2)\|_2 \leq \ell\|w_1 - w_2\|_2$$

  - ρ-Hessian smoothness: The hessian matrix is ρ-Lipschitz, i.e.,

$$\|\nabla^2 f(w_1) - \nabla^2 f(w_2)\|_{sp} \leq \rho\|w_1 - w_2\|_2$$

- $\lambda_{min}(\nabla^2 f(w))$: smallest eigenvalue of the hessian matrix
- α-strongly convex : if $\forall w, \lambda_{min}(\nabla^2 f(w)) \geq \alpha$

Previous results, 1st order

This is a known theorem for strongly convex function.

Thm1, for convex function

f is ℓ-smooth and α-strongly convex. For any $\epsilon > 0$, if run GD with $\eta = 1/\ell$, x_t will be ϵ-close to x^* in iterations

$$\frac{2\ell}{\alpha} \log \frac{\|x_0 - x^*\|}{\epsilon} \quad , x^* \text{ is global minima}$$

Remark :
In general, we no longer have convexity, let alone strong convexity.

ϵ-stationary points

In a more general setting, without "convex" assumption, we analyze convergence to stationary points.

- ϵ-first-order stationary points

$$\|\nabla f(x)\| \leq \epsilon$$

- ϵ-second-order stationary points

$$\|\nabla f(x)\| \leq \epsilon \qquad \lambda_{min}(\nabla^2 f(x)) \geq -\sqrt{\rho\epsilon}$$

Previous results [Nesterov, 1998]

Thm2, for first-order stationary point

f is ℓ-smooth. For any $\epsilon > 0$, if we run GD with $\eta = 1/\ell$, and terminate when $\|\nabla f(x)\| \leq \epsilon$, x_t will be ϵ-first-order stationary point, and terminate within iterations

$$\frac{\ell(f(x_0) - f^*)}{\epsilon^2} \quad , f^* \text{ is global minima}$$

Remark :

1. The iteration complexity doesn't depend on d.
2. a first-order stationary point can be either a local minima or a saddle point or a local maxima. (Not enough)

Strict Saddle Functions

Definition : Strict Saddle

A twice differentiable function $f(x)$ is $(\alpha, \gamma, \epsilon, \zeta)$-strict saddle, if for any x,

- $\|\nabla f(x)\|_2 \geq \epsilon$
- or, $\lambda_{min}(\nabla^2 f(x)) \leq -\gamma \leq 0$
- or, there exists x^* such that $\|x - x^*\|_2 \leq \zeta$, and the function $f(x)$ restricted to 2ζ neighborhood of w^* is α-strongly convex.

Remark :

- Orthogonal tensor decomposition, matrix completion, phase retrieval problem, low rank matrix recovery problems, $\cdots$ are strict saddles.

Strict Saddle Functions

Definition : Strict Saddle

A twice differentiable function $f(x)$ is $(\alpha, \gamma, \epsilon, \zeta)$-strict saddle, if for any x,

1. $\|\nabla f(x)\|_2 \geq \epsilon$
2. or, $\lambda_{min}(\nabla^2 f(x)) \leq -\gamma \leq 0$
3. or, there exists x^* such that $\|x - x^*\|_2 \leq \zeta$, and the function $f(x)$ restricted to 2ζ neighborhood of w^* is α-strongly convex.

Which means :

1. Gradient is large
2. or (stationary point), but we have a negative eigenvalue direction to escape
3. or (stationary point, no negative eigenvalues), but we are pretty close to a local minimum.

Strict Saddle Functions

Definition : Strict Saddle

A twice differentiable function $f(x)$ is $(\alpha, \gamma, \epsilon, \zeta)$-strict saddle, if for any x,

1. $\|\nabla f(x)\|_2 \geq \epsilon$
2. or, $\lambda_{min}(\nabla^2 f(x)) \leq -\gamma \leq 0$
3. or, there exists x^* such that $\|x - x^*\|_2 \leq \zeta$, and the function $f(x)$ restricted to 2ζ neighborhood of w^* is α-strongly convex.

If $\epsilon < \frac{\gamma^2}{\rho}$, then any ϵ-second order stationary point is a local minimum.

Result

(informal) [Ge, 15]

f is ℓ-smooth and ρ-Hessian Lipschitz and $f(x)$ is $(\alpha, \gamma, \epsilon, \zeta)$-strict saddle. With high probability, noisy-SGD will output a point ζ-close to x^* and terminate in $O(poly(d/\epsilon))$.

(informal) [Jin, 17]

f is ℓ-smooth and ρ-Hessian Lipschitz and $f(x)$ is $(\gamma, \epsilon, \zeta)$-strict saddle. With high probability, perturbed-GD will output a point ζ-close to x^*. and terminate in

$$O(\frac{\ell(f(x_0) - f^*)}{\epsilon^2} log^4(\frac{d}{\epsilon^2}))$$

Algorithm [Ge, 15]

Algorithm 1 Noisy Stochastic Gradient

Require: Stochastic gradient oracle $SG(w)$, initial point w_0, desired accuracy κ.

Ensure: w_t that is close to some local minimum $w^\star$.

1: Choose $\eta = \min\{\tilde{O}(\kappa^2 / \log(1/\kappa)), \eta_{\max}\}, T = \tilde{O}(1/\eta^2)$

2: **for** $t = 0$ to $T - 1$ **do**

3: Sample noise n uniformly from unit sphere.

4: $w_{t+1} \leftarrow w_t - \eta(SG(w) + n)$

Remark :

- Add noise every step.
- $w_t = w_{t-1} - \eta(SG(w) + n)$
  $= w_{t-1} - \eta(\nabla f(w_{t-1}) + \xi)$ where $\xi = SG(w_{t-1}) - \nabla f(w_{t-1}) + n$
- Due to the explicitly added noise, $\mathbb{E}\xi\xi^T \succ \frac{1}{d}\mathcal{I}$

Recap

For strict saddle functions, we have

1. large gradient
2. small gradient but a negative eigenvalue
3. close to minima

For 1, gradient suffice to update. For 3, small step size to "fine tune".
$\Rightarrow$ Only in case 2, we have to worry about being stuck.

Algorithm [Jin, 17]

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

$\chi \leftarrow 3\max\{\log(\frac{d\ell\Delta_f}{c\epsilon^2\delta}), 4\},\ \eta \leftarrow \frac{c}{\ell},\ r \leftarrow \frac{\sqrt{c}}{\chi^2}\cdot\frac{\epsilon}{\ell},\ g_{\text{thres}} \leftarrow \frac{\sqrt{c}}{\chi^2}\cdot\epsilon,\ f_{\text{thres}} \leftarrow \frac{c}{\chi^3}\cdot\sqrt{\frac{\epsilon^3}{\rho}},\ t_{\text{thres}}$
$t_{\text{noise}} \leftarrow -t_{\text{thres}} - 1$
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 $\tilde{\mathbf{x}}_{t_{\text{noise}}}$
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta\nabla f(\mathbf{x}_t)$

Algorithm [Jin, 17]

Algorithm 1 Perturbed Gradient Descent (Meta-algorithm)

for $t = 0, 1, \ldots$ **do**
  if perturbation condition holds **then**
    $\mathbf{x}_t \leftarrow \mathbf{x}_t + \xi_t, \qquad \xi_t$ uniformly $\sim \mathbb{B}_0(r)$
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t)$

Perturbation condition :

- If gradient is small & If last perturbation happened at least t_{thres} ago

Stop condition :

- If perturbation happened t_{thres} ago, but no significant progress until now

Algorithm [Jin, 17]

Algorithm 1 Perturbed Gradient Descent (Meta-algorithm)

for $t = 0, 1, \ldots$ **do**
  if perturbation condition holds **then**
    $\mathbf{x}_t \leftarrow \mathbf{x}_t + \xi_t, \qquad \xi_t$ uniformly $\sim \mathbb{B}_0(r)$
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t)$

Remark :

- Not a efficient algorithm since it need full gradient $\nabla f(x_t)$ (noisy SGD need only stochastic gradient)

Proof sketch-overview

For strict saddle functions, we have

1. large gradient $\Rightarrow$ Progress
2. small gradient but a negative eigenvalue $\Rightarrow$ Escape
3. close to minima $\Rightarrow$ Trap

Proof sketch-Progress

For strict saddle functions, we have

1. large gradient $\Rightarrow$ Progress

If f is ℓ-smooth, then for GD with step size $\eta < \frac{1}{\ell}$, we have:

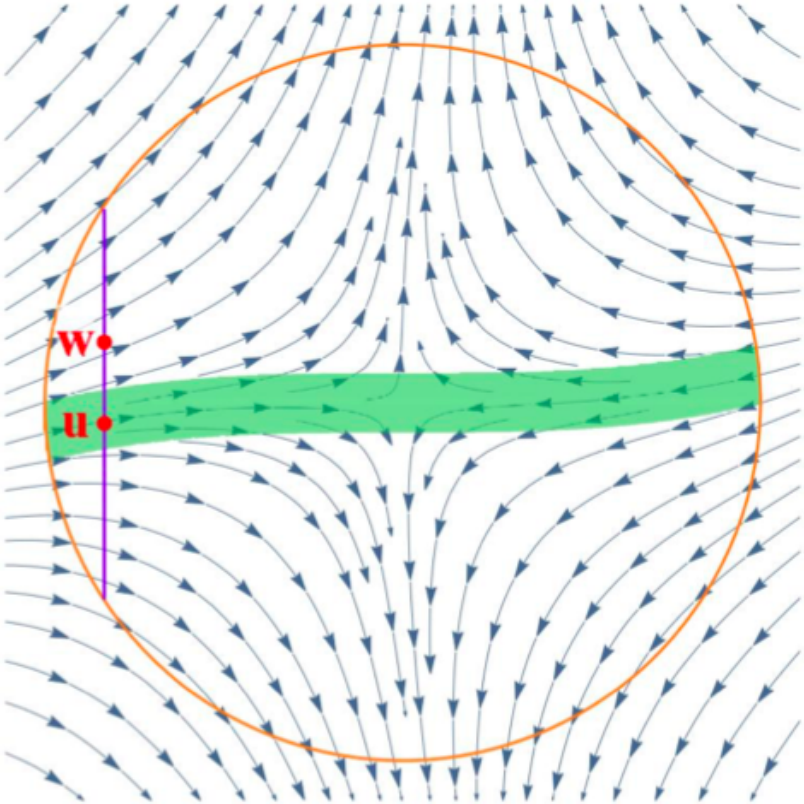
$$f(x_{t+1}) \leq f(x_t) - \frac{\eta}{2}\|\nabla f(x_t)\|^2$$

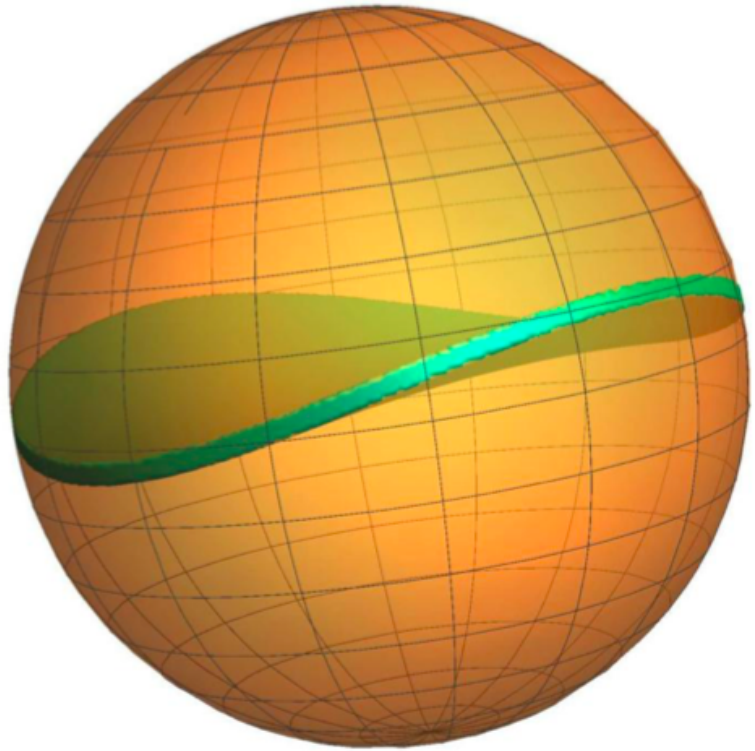
Proof sketch-Trap

When algorithm stop,

- Perturbation happened t_{thres} ago, but f decreased for less than f_{thres}
- Also, at least t_{thres} ago, gradient is small (perturbation condition)
- With high probability, there is no eigenvalue $\leq -\gamma$
- Thus, it's local minima

Proof sketch-Escape





Proof sketch-Escape

After adding the perturbation ($x_0 = x + \xi$), we can view x_0 as coming from a perturbation ball. Furthermore, we can divide perturbation ball into two regions.

- escape_region : which consists of all the points $x \in B_x(r)$ whose f value decreases by at least f_{thres} after t_{thres} steps
- stuck_region : ball - escape_region

Show that $\frac{stuck\_region}{ball}$ is small

Results for today's topic

1. Some problems have nice structure. Ex. Some functions have exponentially many saddle points. However, all local minima are global minima. Furthermore, for strict saddle functions, second-order stationary points are also local minima.
2. There are $poly(d)$-time algorithms that can find a local minimum of strict saddle functions. (by noisy SGD)
3. There are $polylog(d)$-time algorithms that can find a ϵ-second stationary point for functions. (by perturbed GD)
4. Although we do not know the explicit form of saddle points and the stuck region, we know that stuck region must be very "thin". Furthermore, small perturbation can help escape it.

More about SGD

1. Plain SGD can converge to a "2nd-order" stationary point in a number of iterations that is independent of the problem dimension. (by SGD) [2018 ICML]
2. Can escape shallow local minima [2018 ICML]
3. Can find local minima that generalize better than GD (not clear)
4. Jin, 17v2 shows that although [Lee, 16] proved that GD would evantually not being stuck at saddle points, but it can still take exponential time to escape saddle points