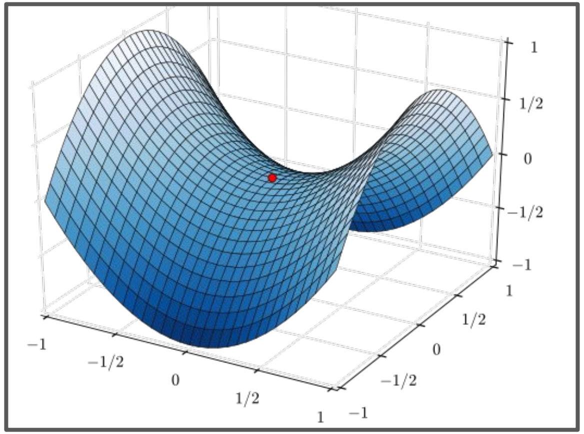
Local Saddle Point Optimization: A Curvature Exploitation Approach

Paper by: Leonard Adolphs, Hadi Daneshmand, Aurelien Lucchi, Thomas Hofmann

Slides by: Paul Vicol

Saddle Point Optimization



- **Goal:** Solve an optimization problem of the form

$$\boxed{\min_{x \in \mathbb{R}^k} \max_{y \in \mathbb{R}^d} f(x, y)}$$

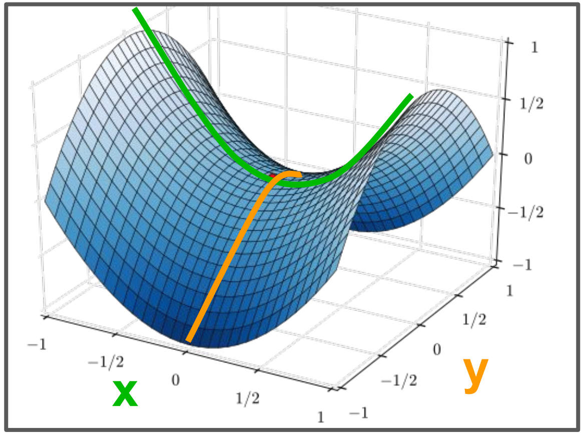
- Where do we encounter such optimization problems?
  - Training *Generative Adversarial Networks (GANs)*

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]$$

  - *Game theory:* “In a two-player zero sum game defined on a continuous space, the equilibrium point is a saddle point.”

Saddle Point Optimization

- An *optimal saddle point* (x^*, y^*) is characterized by:



1. $f(x^*, y) \leq f(x^*, y^*)$ We're at a **max in y** (changing y only gives smaller values of f)

2. $f(x^*, y^*) \leq f(x, y^*)$ We're at a **min in x** (changing x only gives larger values of f)

- The function f is *not necessarily convex in x or concave in y*

➡ We only look for *local saddle points*, where the conditions hold in a *local neighborhood* around (x^*, y^*)

Conditions for Local Optimality

- (x^*, y^*) is a **locally optimal saddle point** on $\mathcal{K}_\gamma^*$ *if and only if*:

$$\nabla f(x^*, y^*) = 0 \qquad \nabla_x^2 f(x^*, y^*) \succ 0 \qquad \nabla_y^2 f(x^*, y^*) \prec 0$$

We're at a *critical/stationary point* & There is *no negative curvature* in the x direction & There is *no positive curvature* in the y direction

Simultaneous Gradient Ascent/Descent

- Classic method: *simultaneous gradient ascent/descent:*

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \end{bmatrix} + \eta \begin{bmatrix} -\nabla_x f(x_t, y_t) \\ \nabla_y f(x_t, y_t) \end{bmatrix}$$

- This method is stable at some *undesired stationary points*
  - **Undesired** = where the function is not a local minimum in x and a maximum in y

Stability

- A *stable stationary point* of an optimization dynamic is a point to which we can converge with non-vanishing probability
- We *would hope* that only the solution of our saddle point problem are the stable stationary points of our optimization scheme

	<u>Minimization</u>	<u>Saddle Point Opt.</u>
Local optimality condition	$\nabla_x^2 f(x) \succ 0$	$\nabla_x^2 f(x,y) \succ 0$ $\nabla_y^2 f(x,y) \prec 0$
Stability condition	$\nabla_x^2 f(x) \succ 0$	$\lambda \begin{bmatrix} -\nabla_x^2 f(x,y) & -\nabla_{xy} f(x,y) \\ \nabla_{yx} f(x,y) & \nabla_y^2 f(x,y) \end{bmatrix}$

➡ Gradient dynamics may introduce additional stable points that are not locally optimal saddle points

Example: GD Converges to Undesired Stable Points

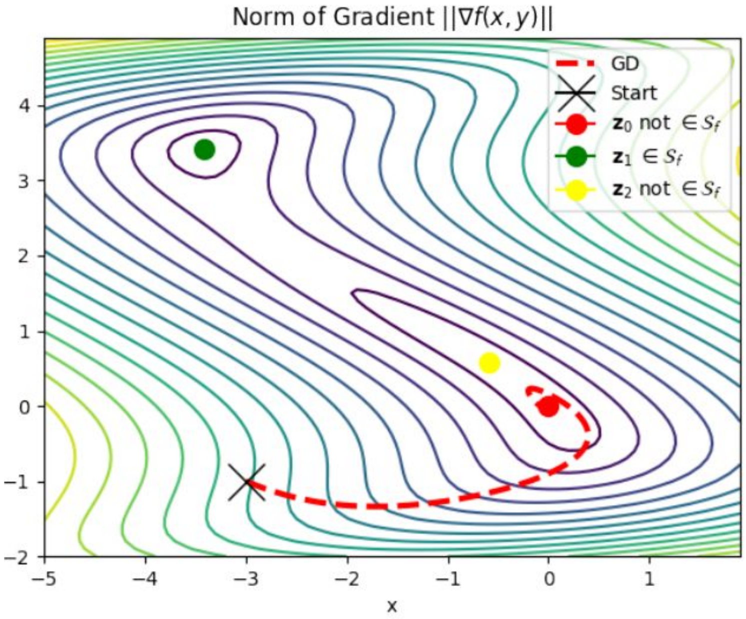
Goal: $$\min_{x\in\mathbb{R}} \max_{y\in\mathbb{R}} \left[f(x,y) = 2x^2 + y^2 + 4xy + \frac{4}{3}y^3 - \frac{1}{4}y^4 \right]$$

Stationary points:

$z_0 = (0,0)$ $\quad$ $z_1 = (-2-\sqrt{2}, 2+\sqrt{2})$ $\quad$ $z_2 = (-2+\sqrt{2}, 2-\sqrt{2})$

$$H(z_0) = \begin{bmatrix} 4 & 4 \\ 4 & 2 \end{bmatrix} \quad H(z_1) = \begin{bmatrix} 4 & 4 \\ 4 & -4\sqrt{2} \end{bmatrix} \quad H(z_2) = \begin{bmatrix} 4 & 4 \\ 4 & 4\sqrt{2} \end{bmatrix}$$

❌ ✓ ❌



Curvature Exploitation for Saddle Point Optimization (CESP)

- How can we *escape from undesired stable points*?
- If we have not yet found a point that is a minimum in x, $\nabla_x^2 f(x,y) \nsucc 0$ so $\nabla_x^2 f(x,y)$ has at least one negative eigenvalue → move along the most negative eigendirection

$$\mathbf{v}_\mathbf{z}^{(-)} = \begin{cases} \frac{\lambda_\theta}{2\rho_\theta}\operatorname{sgn}(\mathbf{v}_\theta^\top \nabla_\theta f(\mathbf{z}))\mathbf{v}_\theta & \text{if } \lambda_\theta < 0 \\ 0 & \text{otherwise} \end{cases}$$

This means that $\nabla_x^2 f(x,y) \nsucc 0$

$$\mathbf{v}_\mathbf{z}^{(+)} = \begin{cases} \frac{\lambda_\varphi}{2\rho_\varphi}\operatorname{sgn}(\mathbf{v}_\varphi^\top \nabla_\varphi f(\mathbf{z}))\mathbf{v}_\varphi & \text{if } \lambda_\varphi > 0 \\ 0 & \text{otherwise} \end{cases}$$

This means that $\nabla_y^2 f(x,y) \nprec 0$

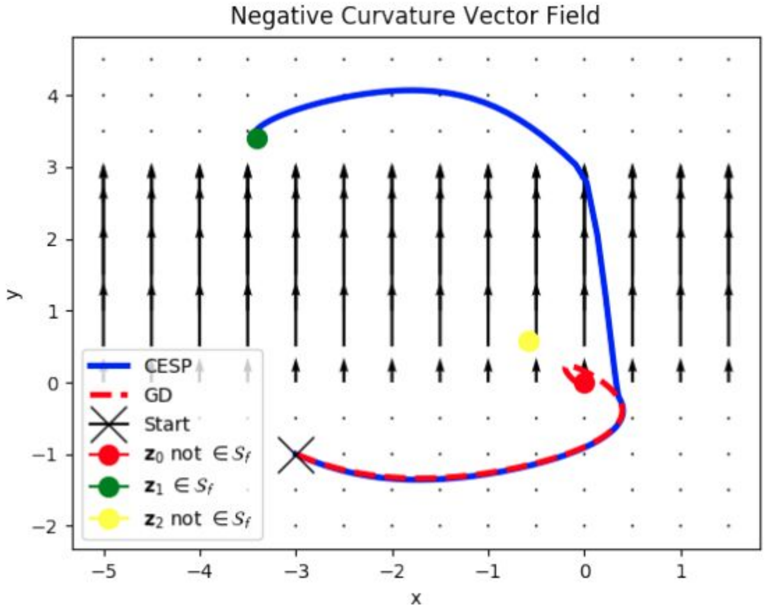
$$\mathbf{v}_\mathbf{z} = (\mathbf{v}_\mathbf{z}^{(-)}, \mathbf{v}_\mathbf{z}^{(+)})$$

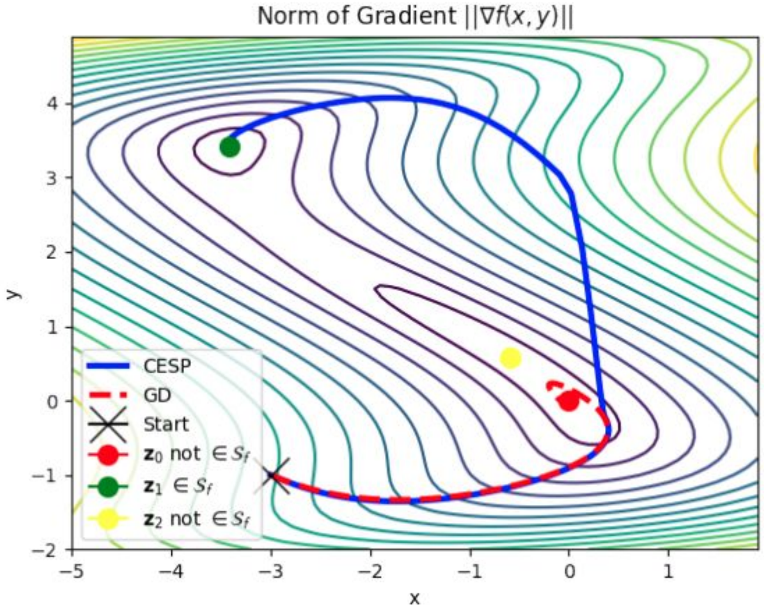
- Modifies simultaneous gradient descent/ascent update with *extreme curvature vector*:

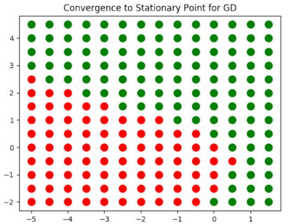
$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \eta \begin{bmatrix} -\nabla_x f(x,y) \\ \nabla_y f(x,y) \end{bmatrix} + \begin{bmatrix} v_z^{(-)} \\ v_z^{(+)} \end{bmatrix}$$

GD & CESP Trajectories

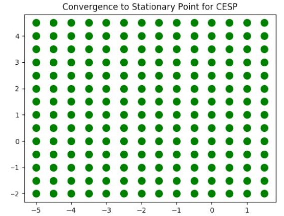
- Comparison of the trajectories of GD and CESP
- The right plot shows the vector field of the extreme curvature. The curvature in the x-dimension is constant and positive, and therefore the extreme curvature is always zero.







GD convergence



CESP convergence

Curvature Exploitation for Linear-Transformed Steps

- They also apply CESP to linearly-transformed gradient steps (in particular Adagrad)

Original linearly-transformed update

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \eta \mathbf{A}_t \begin{bmatrix} -\nabla_x f(x,y) \\ \nabla_y f(x,y) \end{bmatrix}$$

where $\mathbf{A} = \begin{bmatrix} \mathcal{A} & 0 \\ 0 & \mathcal{B} \end{bmatrix}$ is a symmetric, block-diagonal matrix

CESP linearly-transformed update

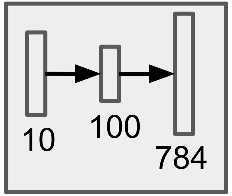
$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \eta \mathbf{A}_t \begin{bmatrix} -\nabla_x f(x,y) \\ \nabla_y f(x,y) \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

where $\mathbf{A}$ must be *positive definite*

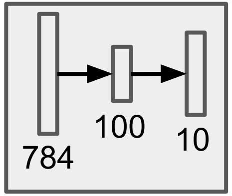
- The set of locally optimal saddle points defined by the simultaneous gradient ascent/descent updates and the set of stable points of the CESP linearly-transformed update *are the same*.

Standard GAN Training

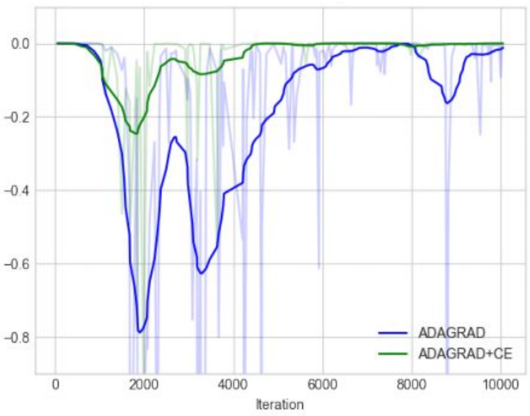
- Train a small GAN on MNIST
- Compare Adagrad to Adagrad w/ curvature exploitation



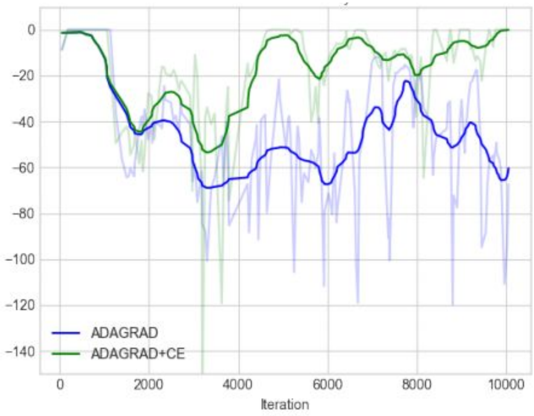
Generator



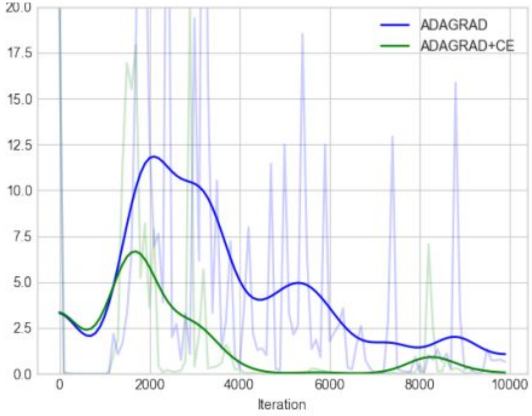
Discriminator



Min Eigenvalue of $\nabla_x^2 f(x, y)$



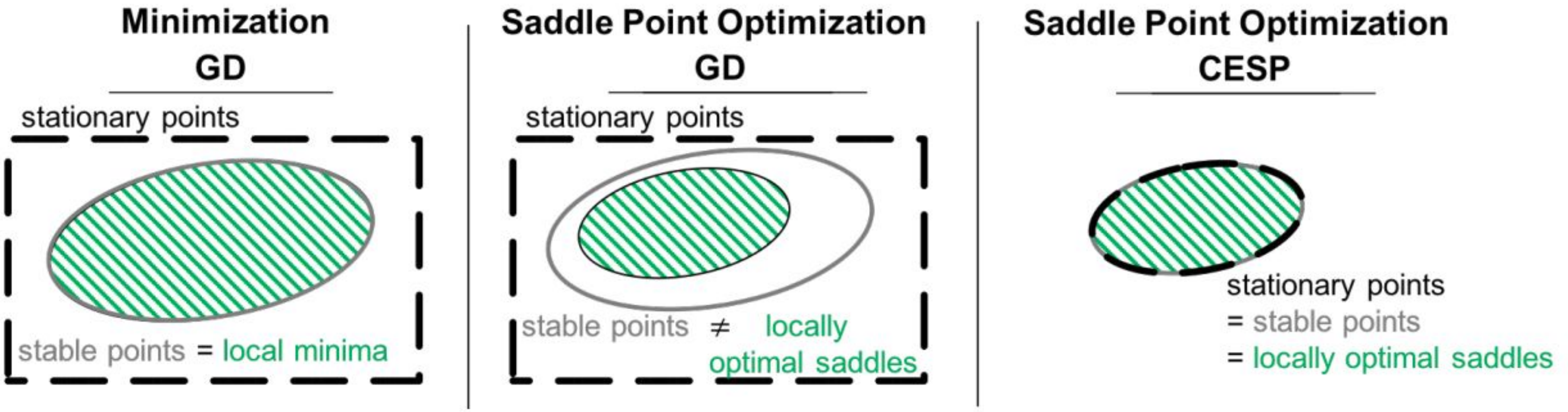
Max Eigenvalue of $\nabla_y^2 f(x, y)$



Squared Gradient Norm

Both methods converge

CESP Guarantees



- CESP provably shrinks the set of stable points to the set of locally optimal solutions

  ➡ Can only converge to locally optimal saddle points

Implementation with Hessian-Vector Products

- Storing and computing the *Hessian in high dimensions is intractable*
  - Need an efficient method to *extract the extreme curvature directions*
- Common approach to obtaining the eigenvector corresponding to the largest absolute eigenvalue of $\nabla_x^2 f(x, y)$ is to run power iterations:

$$v_{t+1} = (\mathbf{I} - \beta \nabla_x^2 f(x, y)) v_t$$

- Can be computed without finding the Hessian, via Hessian-vector products
- Still expensive: How often do we have to compute the extreme curvature?

Summary

- Gradient-based optimization is used for both *minimization* and *saddle-point problems*
- **Problem:** The presence of undesired stable stationary points that are not local optima of the saddle point problem (i.e., minimax problem)

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \end{bmatrix} + \eta \begin{bmatrix} -\nabla_x f(x_t, y_t) \\ \nabla_y f(x_t, y_t) \end{bmatrix}$$

- **Approach:** Exploit curvature information to escape from these undesired stationary points

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \eta \begin{bmatrix} -\nabla_x f(x, y) \\ \nabla_y f(x, y) \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

- *Potentially:* a way to improve GAN training

Q/A