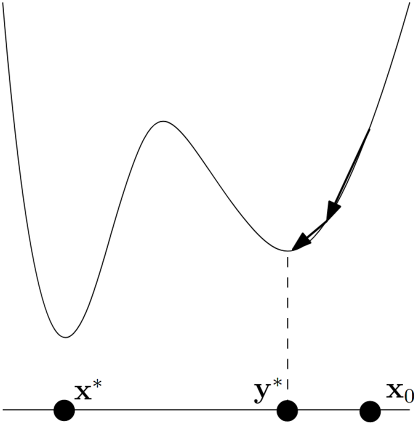
Nonconvex Optimization

Axel Böhm

November 18, 2021

Gradient Descent in the nonconvex world

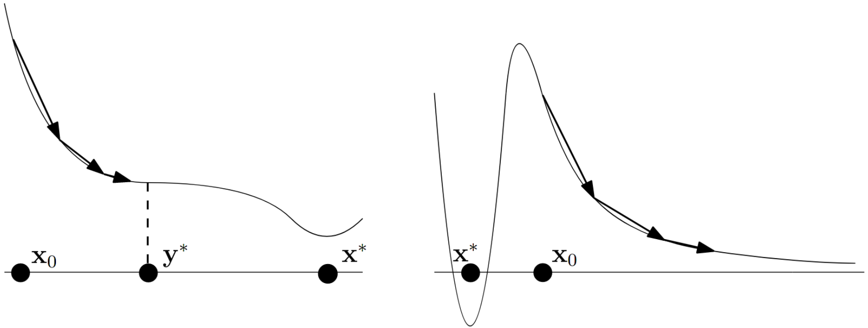
may get stuck in a local minimum and miss the global minimum



Gradient Descent in the nonconvex world II

Even if there is a unique local minimum (equal to the global minimum), we

- may get stuck in a saddle point;
- run off to infinity;
- possibly encounter other bad behaviors.



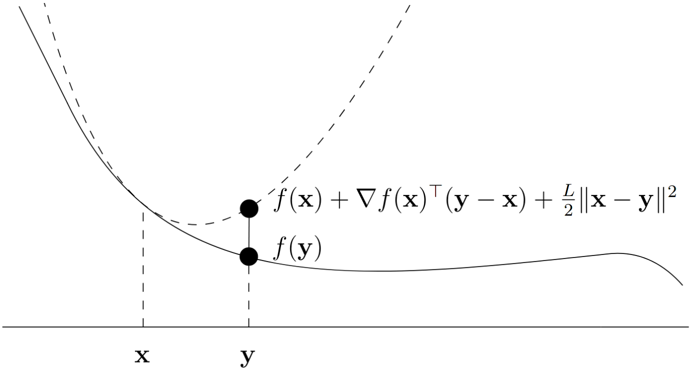
Gradient Descent in the nonconvex world III

- Often, we observe good behavior in practice.
- Theoretical explanations many times missing.
- This lecture: under favorable conditions, we sometimes can say something useful about the behavior of gradient descent, even on nonconvex functions

Smooth (but not necessarily convex) functions

Recall: A differentiable $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth over a convex set X if

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2, \quad \forall x, y \in X.$$



Bounded Hessians $\Rightarrow$ smooth

Lemma

Let $f : \mathbb{R}^d \to \mathbb{R}$ be twice differentiable and

$$\|\nabla^2 f(x)\| \leq L$$

where $\|\cdot\|$ is spectral norm. Then f is L-smooth

Examples:

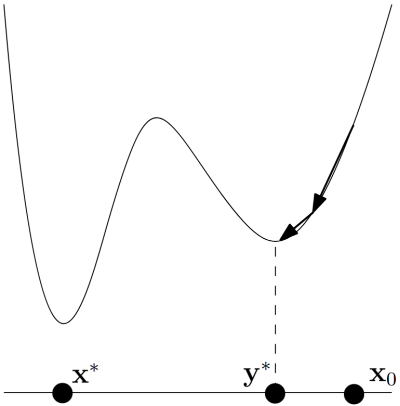
- all quadratic functions $f(x) = x^T A x + b^T x + c$
- $f(x) = \sin(x)$ (many global minima)

Gradient descent on smooth functions

Will prove: $\|\nabla f(x_k)\|^2 \to 0$ for $k \to \infty$ ...

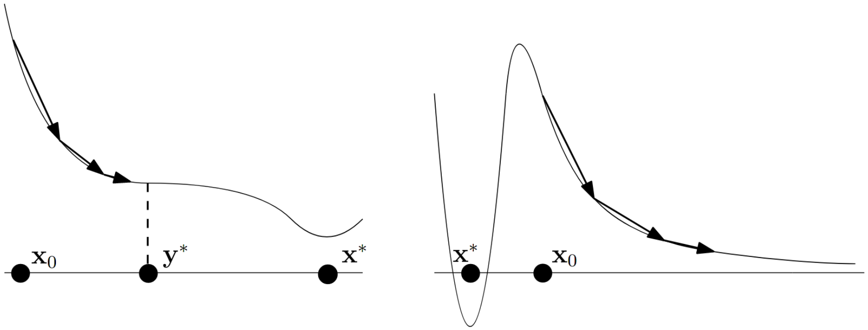
...at the same rate as $f(x_k) - f(x^*) \to 0$ in the convex case.

- $f(x_k) - f(x^*)$ itself may not converge to 0 in the nonconvex case:



What does $\|\nabla f(x_k)\|^2 \to 0$ mean?

- May or may not mean convergence to a critical point $\nabla f(y^*) = 0$
- critical point might not be even local minimum



Figure

Gradient descent on smooth (not necessarily convex) functions

Theorem

Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth with a global minimum x^*. Choosing stepsize $\alpha := \frac{1}{L}$ gradient descent yields

$$\frac{1}{K}\sum_{k=0}^{K-1} \|\nabla f(x_k)\|^2 \leq \frac{2L}{K}(f(x_0) - f(x^*)).$$

In particular, same bound hold for "best" iterate

$$\min_{0\leq k\leq K-1} \|\nabla f(x_k)\|^2 \leq \frac{2L}{K}(f(x_0) - f(x^*))$$

and

$$\lim_{k\to\infty} \|\nabla f(x_k)\|^2 = 0.$$

Gradient descent on smooth functions II: Proof

Smoothness gives:

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2.$$

Use $y = x_{k+1}$ and $x = x_k$

$$f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), -\nabla \alpha f(x_k) \rangle + \frac{L\alpha^2}{2}\|\nabla f(x_k)\|^2$$

to obtain sufficient decrease:

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2.$$

Proof II

sufficient decrease:

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2.$$

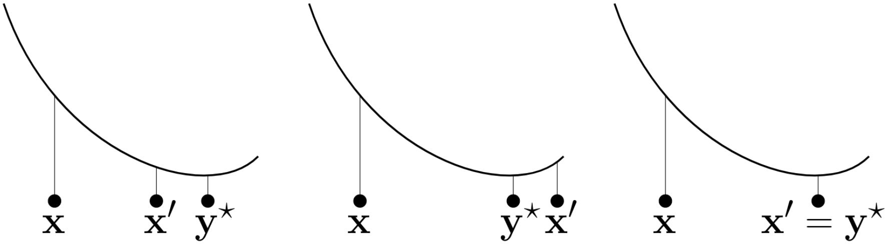
Sum up from $k = 0, 1, \ldots, K-1$ to get

$$\frac{1}{2L}\sum_{k=0}^{K-1}\|\nabla f(x_k)\|^2 \leq f(x_0) - f(x_k) \leq f(x_0) - f(x^*).$$

Multiply by $2L/K$ to get the statement of the theorem.

No overshooting

Under the **smoothness** assumption and appropriate stepsize $\alpha \leq 1/L$, GD cannot pass a critical point:



$\mathbf{x}' = \mathbf{x} - \gamma \nabla f(\mathbf{x}), \gamma < 1/L$ overshooting may happen with $\gamma = 1/L$

Trajectory Analysis

Even if the "landscape" (graph) of a nonconvex function has local minima, saddle points, and flat parts, gradient descent may avoid them and still converge to a global minimum. For this, one needs a good starting point and some theoretical understanding of what happens when we start there—this is trajectory analysis.

Linear models with several outputs

Recall: Learning linear models

- n inputs $x_1, \ldots, x_n$, where each input $x_i \in \mathbb{R}^d$
- n outputs $y_1, \ldots, y_n \in \mathbb{R}$
- Hypothesis (after centering / no bias):

$$y_i \approx w^T x_i,$$

for a weight vector $\boldsymbol{w} = (w_1, ..., w_d) \in \mathbb{R}^d$ to be learned.

Now more than one output value:

- n outputs $y_1, ..., y_n$, where each output $y_i \in \mathbb{R}^m$
- Hypothesis:

$$y_i \approx W x_i,$$

for a weight matrix $W \in \mathbb{R} m \times d$ to be learned.

Minimizing the least squares error

Compute

$$W^* = \arg\min_{W \in \mathbb{R}^{m \times d}} \sum_{i=1}^{n} \|Wx_i - y_i\|^2.$$

- $X \in \mathbb{R}^{d \times n}$: matrix whose columns are the x_i
- $Y \in \mathbb{R}^{m \times n}$: matrix whose columns are the y_i

Then

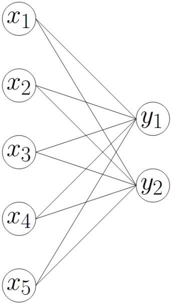
$$W^* = \arg\min_{W \in \mathbb{R}^{m \times d}} \sum_{i=1}^{n} \|Wx_i - y_i\|^2.$$

where $\|A\|_F = \sqrt{\sum_{i,j} a_{i,j}}$ is the Frobenius norm of a matrix A. Frobenius norm of A = Euclidean norm of `vec(A)` ("flattening" of A).

Minimizing the least squares error II

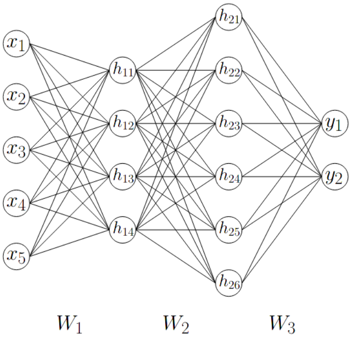
$$W^* = \underset{W \in \mathbb{R}^{m \times d}}{\arg\min} \|WX - Y\|_F^2$$

is the global minimum of a convex quadratic function $f(W)$.
To find W^*, solve $\nabla f(W) = 0$ (system of linear equations)
$\Leftrightarrow$ training a linear neural network with one layer under least squares loss.



$$\mathbf{x} \mapsto \mathbf{y} = W\mathbf{x}$$

Deep linear neural networks



$$\mathbf{x} \mapsto \mathbf{y} = W_3 W_2 W_1 \mathbf{x}$$

Not more expressive:

$$x \mapsto W_3 W_2 W_1 x \Leftrightarrow x \mapsto Wx, \quad \text{for } W := W_3 W_2 W_1$$

Training deep linear neural networks

With ℓ layers:

$$W^* = \underset{W_1, W_2, \ldots, W_\ell}{\arg\min} \|W_1 W_2 \cdots W_\ell X - Y\|_F^2$$

Nonconvex function for $\ell > 1$.
Playground to understand why training deep neural networks with gradient descent works.
Here: all matrices are 1×1, $W_i = x_i$, $X = 1$, $Y = 1$, $\ell = d$
$\Rightarrow f : \mathbb{R}^d \to \mathbb{R}$,

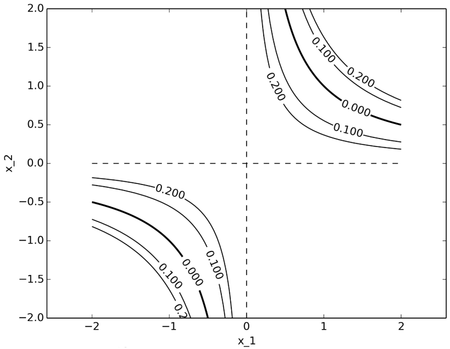
$$f(x) := \frac{1}{2}\left(\prod_{j=1}^{d} x_j - 1\right)^2.$$

Toy example in our simple playground

A simple nonconvex function

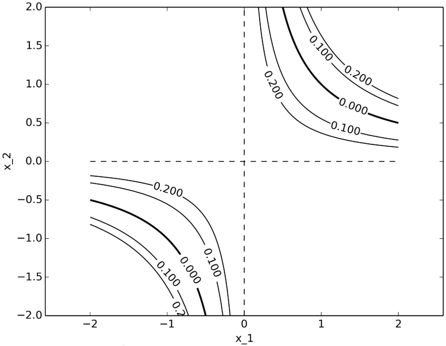
Nonconvex level sets: $f(x) = \frac{1}{2}\left(\prod_j x_j\right)$.

Dimensions is fixed so we ignore it



Gradient and critical points

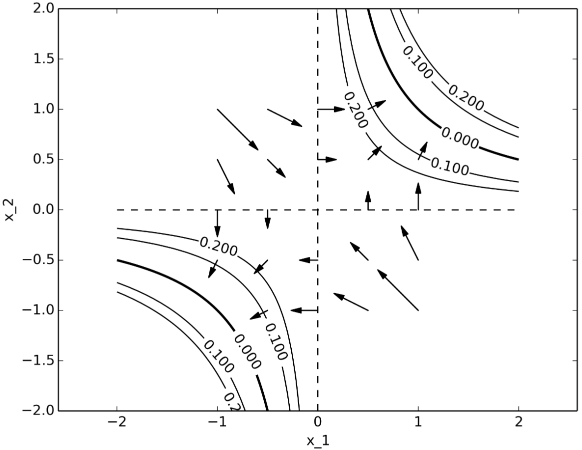
$$\nabla f(x) = \left(\prod_j x_j\right)\left(\prod_{j\neq 1} x_j, \cdots, \prod_{j\neq d} x_j\right).$$



Critical points ($\nabla f(x) = 0$) are either:

- global minima: if $\prod_j x_j = 1$
  - $d = 2$: hyperbola
- saddle point: if at least two of x_j are zero
  - $d = 2$: only the origin $(0, 0)$

Negative gradient directions



Convergence to global minimum from almost everywhere.

Convergence analysis: Overview

Convergence of GD holds for any $d > 1$ and from anywhere in $X = \{x : x > 0, \prod_j x_j \leq 1\}$.

- f is not smooth over X. But is smooth along the trajectory: For suitable L we still get

$$f(x_{k+1}) = f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2. \tag{SD}$$

- saddle points have (at least two) zero entries $\Rightarrow$ function value $\geq 1/2$.
- any starting $x_0 \in X$ has $f(x_0) < 1/2$
- cannot converge to saddle points through (SD)

Still does not imply convergence to global minimum:

- Sublevel sets are unbounded: GD can run off to ∞

Convergence analysis: Overview II

For $x > 0, \prod_j x_j \geq 1$, we can also show convergence:
$\Rightarrow$ convergence anywhere in the interior of the positive orthant $\{x : x > 0\}$.
For this, recall that

$$\nabla f(x) = \left(\prod_j x_j\right)\left(\prod_{j\neq 1} x_j, \cdots, \prod_{j \neq d} x_j\right).$$

- if $\prod_j x_j \geq 1$ then $\nabla f(x) \geq 0$
- which implies that $x_1 \leq x_0$ (componentwise)
- iterates remain in a bounded set $\Rightarrow$ smoothness on this set

Definition

Let $x > 0$ (componentwise), and let $c \geq 1$. x is called c-balanced if $x_i \leq c x_j$ for all $1 \leq i, j \leq d$.

Theorem

Let $c \geq 1$ and $\delta > 0$ such that $x^0 > 0$ is c-balanced with $\delta \leq \prod_j x_j^0 < 1$. Choosing the stepsize

$$\gamma = \frac{1}{3dc^2}$$

gradient descent satisfies

$$f(x^k) \leq \left(1 - \frac{\delta^2}{3c^4}\right) f(x^0).$$

Discussion

- Error converges to 0 exponentially fast.
- But there's a catch: Consider $x^0 = (1/2, \dots, 1/2)$. Then $\delta \leq \prod_j x_j^0 = 2^{-d}$
- Decrease in function value per step by factor

$$\left(1 - \frac{1}{34^d}\right).$$

- Contraction coefficient depends exponentially bad on dimension
- for polynomial runtime must start at distance $\mathcal{O}(1/\sqrt{d})$ from optimality.

Matrix completion

Matrix completion is the problem of recovering a **low rank** ($r \ll d$) matrix $M^{d\times d}$ from **partially observed** entries:

Application: Netflix problem

$$\min_{X\in\mathbb{R}^{d\times d}} \quad \text{rank}(X)$$
$$\text{subject to } X_{i,j} = M_{i,j}, \quad \forall i,j \in \Omega$$

But rank is **not continuous**. . .

Convex matrix completion

Typically **convex** reformulations are considered via the **Nuclear norm** (sum of singular values)

$$\min_{X \in \mathbb{R}^{d \times d}} \|X\|_* := \sum_j \sigma_j(X)$$

$$\text{subject to } X_{i,j} = M_{i,j}, \quad \forall i,j \in \Omega$$

- strong theoretical guarantees
- can be expensive
  - $\mathcal{O}(d^3)$ running time
  - $\mathcal{O}(d^2)$ memory.

Nonconvex matrix completion

Can be cast in the bilinear $X \approx UV^T$ form which gives:

$$\min_{U,V \in \mathbb{R}^{d\times r}} \sum_{i,j\in\Omega} \|(UV^T)_{i,j} - M_{j,j}\|^2.$$

- many global minima
- if $UV^T = M$ then $(UQ)(VQ)^T = M$ for any orthonormal matrix Q
  orthonormal: $QQ^T = \mathrm{Id}$

No spurious local minima!

Can often be efficiently solved by GD or alternating minimization.