Optimization for Machine Learning
CS-439

Lecture 7: Non-convex opt., Newton's Method

Martin Jaggi

EPFL – `github.com/epfml/OptML_course`

April 5, 2019

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**.

2018: trajectory analysis for training deep linear linear neural networks, under suitable conditions [ACGH18].

Here: vastly simplified setting that allows us to show the main ideas (and limitations).

Linear models with several outputs

Recall: Learning linear models

- n inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n$, where each input $\mathbf{x}_i \in \mathbb{R}^d$
- n outputs $y_1, \ldots, y_n \in \mathbb{R}$
- Hypothesis (after centering):

$$y_i \approx \mathbf{w}^\top x_i,$$

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

Now more than one output value:

- n outputs $\mathbf{y}_1, \ldots, \mathbf{y}_n$, where each output $\mathbf{y}_i \in \mathbb{R}^m$
- Hypothesis:

$$\mathbf{y}_i \approx W\mathbf{x}_i,$$

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

Minimizing the least squares error

Compute

$$W^{\star} = \underset{W \in \mathbb{R}^{m \times d}}{\operatorname{argmin}} \sum_{i=1}^{n} \|W\mathbf{x}_i - \mathbf{y}_i\|^2 .$$

- $X \in \mathbb{R}^{d \times n}$: matrix whose columns are the $\mathbf{x}_i$
- $Y \in \mathbb{R}^{m \times n}$: matrix whose columns are the $\mathbf{y}_i$

Then

$$W^{\star} = \underset{W \in \mathbb{R}^{m \times d}}{\operatorname{argmin}} \|WX - Y\|_F^2,$$

where $\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$ is the Frobenius norm of a matrix A.

Frobenius norm of A = Euclidean norm of $\operatorname{vec}(A)$ ("flattening" of A)

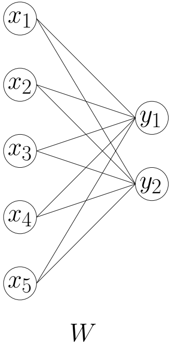
Minimizing the least squares error II

$$W^\star = \underset{W \in \mathbb{R}^{m \times d}}{\operatorname{argmin}} \|WX - Y\|_F^2$$

is the global minimum of a convex quadratic function $f(W)$.

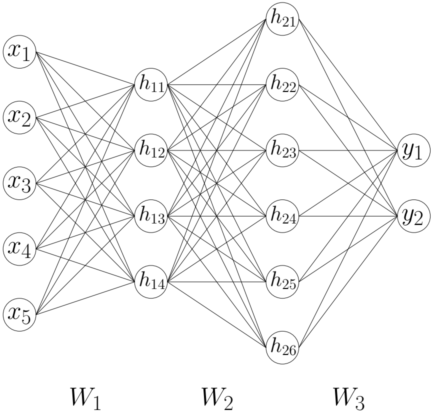
To find $W^\star$, solve $\nabla f(W) = \mathbf{0}$ (system of linear equations).

$\Leftrightarrow$ training a **linear neural network with one layer** under least squares error.



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

Deep linear neural networks



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

Not more expressive:

$$\mathbf{x} \mapsto \mathbf{y} = W_3 W_2 W_1 \mathbf{x} \quad \Leftrightarrow \quad \mathbf{x} \mapsto \mathbf{y} = W\mathbf{x},\ W := W_3 W_2 W_1.$$

Training deep linear neural networks

With ℓ layers:

$$W^\star = \underset{W_1, W_2, \ldots, W_\ell}{\operatorname{argmin}} \|W_\ell W_{\ell-1} \cdots W_1 X - Y\|_F^2,$$

Nonconvex function for $\ell > 1$.

Simple playground in which we can try to understand why training deep neural networks with gradient descent works.

Here: all matrices are 1×1, $W_i = x_i, X = 1, Y = 1, \ell = d \quad \Rightarrow f : \mathbb{R}^d \to \mathbb{R}$,

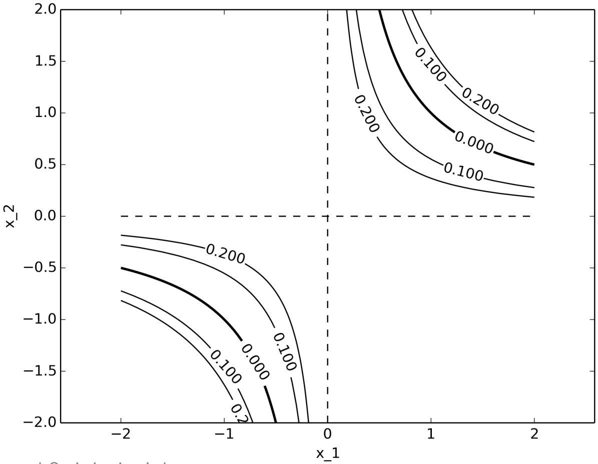
$$f(\mathbf{x}) := \frac{1}{2}\left(\prod_{k=1}^{d} x_k - 1\right)^2.$$

Toy example in our simple playground.

But analysis of gradient descent on f has similar ingredients as the one on general deep linear neural networks [ACGH18].

A simple nonconvex function

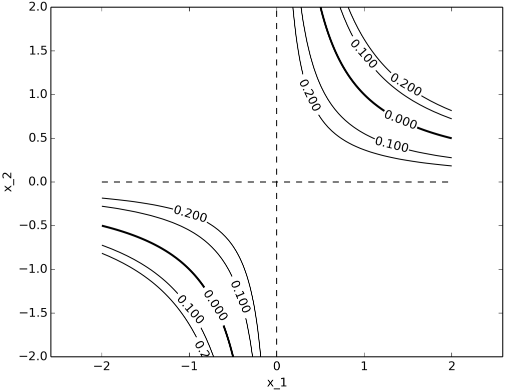
As d is fixed, abbreviate $\prod_{k=1}^{d} x_k$ by $\prod_k x_k$: $f(\mathbf{x}) = \frac{1}{2}\left(\prod_k x_k - 1\right)^2$



Level set plot

The gradient

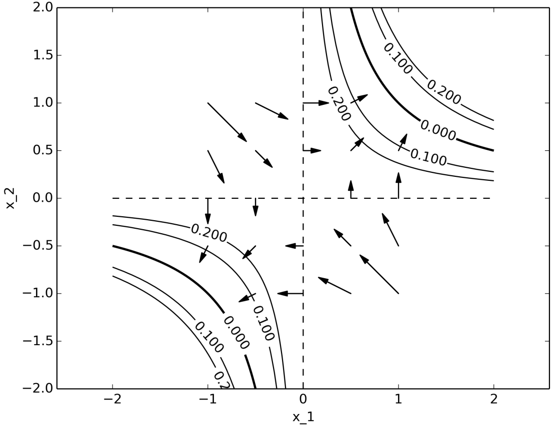
$$\nabla f(\mathbf{x}) = \left(\prod_k x_k - 1\right)\left(\prod_{k\neq 1} x_k, \ldots, \prod_{k\neq d} x_k\right).$$



Critical points ($\nabla f(\mathbf{x}) = \mathbf{0}$):

- $\prod_k x_k = 1$ (global minima)
  - $d = 2$: the hyperbola $\{(x_1, x_2) : x_1 x_2 = 1\}$
- at least two of the x_k are zero (saddle points)
  - $d = 2$: the origin $(x_1, x_2) = (0, 0)$

Negative gradient directions (followed by gradient descent)



Difficult to avoid convergence to a global minimum, but it is possible (Exercise 37).

Convergence analysis: Overview

Want to show that for any $d > 1$, and from anywhere in $X = \{\mathbf{x} : \mathbf{x} > \mathbf{0}, \prod_k \mathbf{x}_k \leq 1\}$, gradient descent will converge to a global minimum.

f is not smooth over X. We show that f is smooth along the trajectory of gradient descent for suitable L, so that we get sufficient decrease

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|^2, \quad t \geq 0.$$

Then, we cannot converge to a saddle point: all these have (at least two) zero entries and therefore function value $1/2$. But for starting point $\mathbf{x}_0 \in X$, we have $f(\mathbf{x}_0) < 1/2$, so we can never reach a saddle while decreasing f.

Doesn't this imply converge to a global mimimum? No!

- Sublevel sets are unbounded, so we could in principle run off to infinity.
- Other bad things might happen (we haven't characterized what can go wrong).

Convergence analysis: Overview II

For $\mathbf{x} > \mathbf{0}, \prod_k \mathbf{x}_k \geq 1$, we also get convergence (Exercise 36).

$\Rightarrow$ convergence from anywhere in the interior of the positive orthant $\{\mathbf{x} : \mathbf{x} > \mathbf{0}\}$.

But there are also starting points from which gradient descent will not converge to a global minimum (Exercise 37).

Main tool: Balanced iterates

Definition

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

Any initial iterate $\mathbf{x}_0 > \mathbf{0}$ is c-balanced for some (possibly large) c.

Lemma

Let $\mathbf{x} > \mathbf{0}$ be c-balanced with $\prod_k x_k \leq 1$. Then for any stepsize $\gamma > 0$, $\mathbf{x}' := \mathbf{x} - \gamma \nabla f(\mathbf{x})$ satisfies $\mathbf{x}' \geq \mathbf{x}$ (componentwise) and is also c-balanced.

Proof.

$\Delta := -\gamma(\prod_k x_k - 1)(\prod_k x_k) \geq 0. \qquad \nabla f(\mathbf{x}) = (\prod_k x_k - 1)\left(\prod_{k\neq 1} x_k, \ldots, \prod_{k \neq d} x_k\right).$

Gradient descent step:

$$x'_k = x_k + \frac{\Delta}{x_k} \geq x_k, \quad k = 1, \ldots, d.$$

For i, j, we have $x_i \leq c x_j$ and $x_j \leq c x_i$ ($\Leftrightarrow 1/x_i \leq c/x_j$). We therefore get

$$x'_i = x_i + \frac{\Delta}{x_i} \leq c x_j + \frac{\Delta c}{x_j} = c x'_j.$$

□

Bounded Hessians along the trajectory

Compute $\nabla^2 f(\mathbf{x})$:

$\nabla^2 f(\mathbf{x})_{ij}$ is the j-th partial derivative of the i-th entry of $\nabla f(\mathbf{x})$.

$$(\nabla f)_i = \left(\prod_k x_k - 1\right) \prod_{k \neq i} x_k$$

$$\nabla^2 f(\mathbf{x})_{ij} = \begin{cases} \left(\prod_{k \neq i} x_k\right)^2, & j = i \\ 2 \prod_{k \neq i} x_k \prod_{k \neq j} x_k - \prod_{k \neq i,j} x_k, & j \neq i \end{cases}$$

Need to bound $\prod_{k \neq i} x_k, \quad \prod_{k \neq j} x_k, \quad \prod_{k \neq i,j} x_k$!

Bounded Hessians along the trajectory II

Lemma

Suppose that $\mathbf{x} > \mathbf{0}$ *is* c*-balanced. Then for any* $I \subseteq \{1, \ldots, d\}$*, we have*

$$\left(\frac{1}{c}\right)^{|I|} \left(\prod_k x_k\right)^{1-|I|/d} \leq \prod_{k \notin I} x_k \leq c^{|I|} \left(\prod_k x_k\right)^{1-|I|/d}.$$

Proof.

For any i, we have $x_i^d \geq (1/c)^d \prod_k x_k$ by balancedness, hence $x_i \geq (1/c)(\prod_k x_k)^{1/d}$.It follows that

$$\prod_{k \notin I} x_k = \frac{\prod_k x_k}{\prod_{i \in I} x_i} \leq \frac{\prod_k x_k}{(1/c)^{|I|}(\prod_k x_k)^{|I|/d}} = c^{|I|} \left(\prod_k x_k\right)^{1-|I|/d}.$$

The lower bound follows in the same way from $x_i^d \leq c^d \prod_k x_k$. $\square$

Bounded Hessians along the trajectory III

Lemma

Let $\mathbf{x} > \mathbf{0}$ *be* c*-balanced with* $\prod_k x_k \leq 1$*. Then*

$$\left\|\nabla^2 f(\mathbf{x})\right\| \leq \left\|\nabla^2 f(\mathbf{x})\right\|_F \leq 3dc^2.$$

where $\|A\|_F$ *is the Frobenius norm and* $\|A\|$ *the spectral norm.*

Proof.

$\|A\| \leq \|A\|_F$: Exercise 38. Now use previous lemma and $\prod_k x_k \leq 1$:

$$\left|\nabla^2 f(\mathbf{x})_{ii}\right| = |(\prod_{k \neq i} x_k)^2| \leq c^2$$

$$\left|\nabla^2 f(\mathbf{x})_{ij}\right| \leq |2 \prod_{k \neq i} x_k \prod_{k \neq j} x_k| + |\prod_{k \neq i,j} x_k| \leq 3c^2.$$

Hence, $\left\|\nabla^2 f(\mathbf{x})\right\|_F^2 \leq 9d^2c^4$. Taking square roots, the statement follows. $\square$

Smoothness along the trajectory

Lemma

Let $\mathbf{x} > \mathbf{0}$ be c-balanced with $\prod_k x_k < 1$, $L = 3dc^2$. Let $\gamma := 1/L$. Then for all $0 \leq \nu \leq \gamma$,

$$\mathbf{x}' := \mathbf{x} - \nu \nabla f(\mathbf{x}) \geq \mathbf{x}$$

is c-balanced with $\prod_k x'_k \leq 1$, and f is smooth with parameter L over the line segment connecting $\mathbf{x}$ and $\mathbf{x} - \gamma \nabla f(\mathbf{x})$.

Proof.

- $\mathbf{x}' \geq \mathbf{x} > \mathbf{0}$ is c-balanced by Lemma 6.5.
- $\nabla f(\mathbf{x}) \neq \mathbf{0}$ (due to $\mathbf{x}' > \mathbf{0}, \prod_k x_k < 1$, we can't be at a critical point).
- No overshooting: we can't reach $\prod_k x'_k = 1$ (global minimum) for $\nu < \gamma$, as f is smooth with parameter L between $\mathbf{x}$ and $\mathbf{x}'$ (using previous bound on Hessians in Lemma 6.1).
- By continutity, $\prod_k x'_k \leq 1$ for all $\nu \leq \gamma$.
- f is smooth with parameter L between $\mathbf{x}$ and $\mathbf{x}'$ for $\nu = \gamma$.

Convergence

Theorem

Let $c \geq 1$ *and* $\delta > 0$ *such that* $\mathbf{x}_0 > \mathbf{0}$ *is* c*-balanced with* $\delta \leq \prod_k (\mathbf{x}_0)_k < 1$*. Choosing stepsize*

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

gradient descent satisfies

$$f(\mathbf{x}_T) \leq \left(1 - \frac{\delta^2}{3c^4}\right)^T f(\mathbf{x}_0), \quad T \geq 0.$$

- Error converges to 0 exponentially fast.
- Exercise 39: iterates themselves converge (to an optimal solution).

Convergence: Proof

Proof.

- For $t \geq 0$, f is smooth between $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ with parameter $L = 3dc^2$.
- Sufficient decrease:

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{6dc^2} \left\| \nabla f(\mathbf{x}_t) \right\|^2.$$

For every c-balanced $\mathbf{x}$ with $\delta \leq \prod_k x_k \leq 1$, $\left\| \nabla f(\mathbf{x}) \right\|^2$ equals

$$2f(\mathbf{x}) \sum_{i=1}^{d} \left(\prod_{k \neq i} x_k \right)^2 \geq 2f(\mathbf{x}) \frac{d}{c^2} \left(\prod_k x_k \right)^{2-2/d} \geq 2f(\mathbf{x}) \frac{d}{c^2} \left(\prod_k x_k \right)^2 \geq 2f(\mathbf{x}) \frac{d}{c^2} \delta^2.$$

- Hence, $f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{6dc^2} 2f(\mathbf{x}_t) \frac{d}{c^2} \delta^2 = f(\mathbf{x}_t) \left(1 - \frac{\delta^2}{3c^4} \right).$

Discussion

Fast convergence as for strongly convex functions!

But there is a catch...

Consider starting solution $\mathbf{x}_0 = (1/2, \ldots, 1/2)$.

$\delta \leq \prod_k (\mathbf{x}_0)_k = 2^{-d}$.

Decrease in function value by a factor of

$$\left(1 - \frac{1}{3 \cdot 4^d}\right),$$

per step.

Need $T \approx 4^d$ to reduce the initial error by a constant factor not depending on d.

Problem: gradients are exponentially small in the beginning, extremely slow progress.

For polynomial runtime, must start at distance $O(1/\sqrt{d})$ from optimality.

Chapter 7

Newton's Method

1-dimensional case: Newton-Raphson method

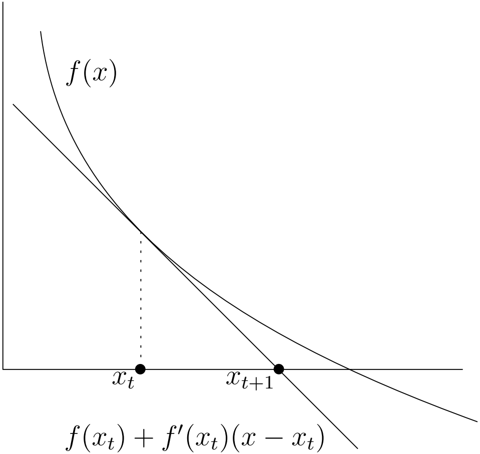
Goal: find a zero of differentiable $f : \mathbb{R} \to \mathbb{R}$.

Method:

$$x_{t+1} := x_t - \frac{f(x_t)}{f'(x_t)}, \quad t \geq 0.$$

x_{t+1} solves

$$f(x_t) + f'(x_t)(x - x_t) = 0,$$



The Babylonian method

Computing square roots: find a zero of $f(x) = x^2 - R, R \in \mathbb{R}_+$.

Newton-Raphson step:

$$x_{t+1} = x_t - \frac{f(x_t)}{f'(x_t)} = x_t - \frac{x_t^2 - R}{2x_t} = \frac{1}{2}\left(x_t + \frac{R}{x_t}\right).$$

Starting from $x_0 > 0$, we have

$$x_{t+1} = \frac{1}{2}\left(x_t + \frac{R}{x_t}\right) \geq \frac{x_t}{2}.$$

Starting from $x_0 = R \geq 1$, it takes $O(\log R)$ steps to get $x_t - \sqrt{R} < 1/2$ (Exercise 40).

The Babylonian method - Takeoff

Suppose $x_0 - \sqrt{R} < 1/2$ (achievable after $O(\log R)$ steps).

$$x_{t+1} - \sqrt{R} = \frac{1}{2}\left(x_t + \frac{R}{x_t}\right) - \sqrt{R} = \frac{x_t}{2} + \frac{R}{2x_t} - \sqrt{R} = \frac{1}{2x_t}\left(x_t - \sqrt{R}\right)^2.$$

Assume $R \geq 1/4$.Then all iterates have value at least $\sqrt{R} \geq 1/2$. Hence we get

$$x_{t+1} - \sqrt{R} \leq \left(x_t - \sqrt{R}\right)^2.$$

$$x_T - \sqrt{R} \leq \left(x_0 - \sqrt{R}\right)^{2^T} < \left(\frac{1}{2}\right)^{2^T}, \quad T \geq 0.$$

To get $x_T - \sqrt{R} < \varepsilon$, we only need $T = \log\log(\frac{1}{\varepsilon})$ steps!

The Babylonian method - Example

$R = 1000$, IEEE 754 double arithmetic

- 7 steps to get $x_7 - \sqrt{1000} < 1/2$
- 3 more steps to get x_{10} equal to $\sqrt{1000}$ up to machine precision (53 binary digits).
- First phase: $\approx$ one more correct digit per iteration
- Last phase, $\approx$ double the number of correct digits in each iteration!

Once you're close, you're there...

Newton's method for optimization

1-dimensional case: Find a global minimum $x^\star$ of a differentiable convex function $f : \mathbb{R} \to \mathbb{R}$.

Can equivalently search for a zero of the derivative f': Apply the Newton-Raphson method to f'.

Update step:

$$x_{t+1} := x_t - \frac{f'(x_t)}{f''(x_t)} = x_t - f''(x_t)^{-1} f'(x_t)$$

(needs f twice differentiable).

d-dimensional case: Newton's method for minimizing a convex function $f : \mathbb{R}^d \to \mathbb{R}$:

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \nabla^2 f(\mathbf{x}_t)^{-1} \nabla f(\mathbf{x}_t)$$

Newton's method = adaptive gradient descent

General update scheme:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - H(\mathbf{x}_t)\nabla f(\mathbf{x}_t),$$

where $H(\mathbf{x}) \in \mathbb{R}^{d\times d}$ is some matrix.

Newton's method: $H = \nabla^2 f(\mathbf{x}_t)^{-1}$.

Gradient descent: $H = \gamma I$.

Newton's method: "adaptive gradient descent", adaptation is w.r.t. the local geometry of the function at $\mathbf{x}_t$.

Convergence in one step on quadratic functions

A nondegenerate quadratic function is a function of the form

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top M\mathbf{x} - \mathbf{q}^\top\mathbf{x} + c,$$

where $M \in \mathbb{R}^{d\times d}$ is an invertible symmetric matrix, $\mathbf{q} \in \mathbb{R}^d, c \in R$. Let $\mathbf{x}^\star = M^{-1}\mathbf{q}$ be the unique solution of $\nabla f(\mathbf{x}) = \mathbf{0}$.

- $\mathbf{x}^\star$ is the unique global minimum if f is convex.

Lemma

On nondegenerate quadratic functions, with any starting point $\mathbf{x}_0 \in \mathbb{R}^d$, Newton's method yields $\mathbf{x}_1 = \mathbf{x}^\star$.

Proof.

We have $\nabla f(\mathbf{x}) = M\mathbf{x} - \mathbf{q}$ (this implies $\mathbf{x}^\star = M^{-1}\mathbf{q}$) and $\nabla^2 f(\mathbf{x}) = M$. Hence,

$$\mathbf{x}_1 = \mathbf{x}_0 - \nabla^2 f(\mathbf{x}_0)^{-1}\nabla f(\mathbf{x}_0) = \mathbf{x}_0 - M^{-1}(M\mathbf{x}_0 - \mathbf{q}) = M^{-1}\mathbf{q} = \mathbf{x}^\star.$$

$\square$

Bibliography

Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu.
A convergence analysis of gradient descent for deep linear neural networks.
CoRR, abs/1810.02281, 2018.