Chapter 11: Optimization

Learning Objectives

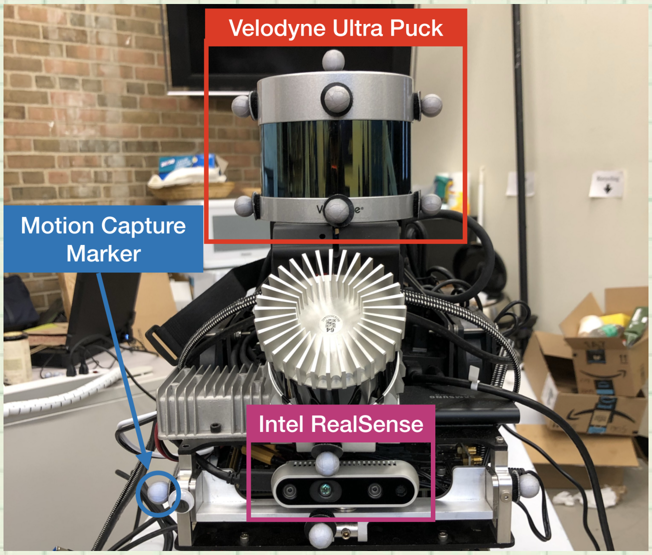
- Mathematics is used to describe physical phenomena, pose engineering problems, and solve engineering problems. We close our Y1-introduction to Computational Linear Algebra by showing how linear algebra and computation allow you to use a notion of "optimality" as a criterion for selecting among a set of solutions to an engineering problem.

Outcomes

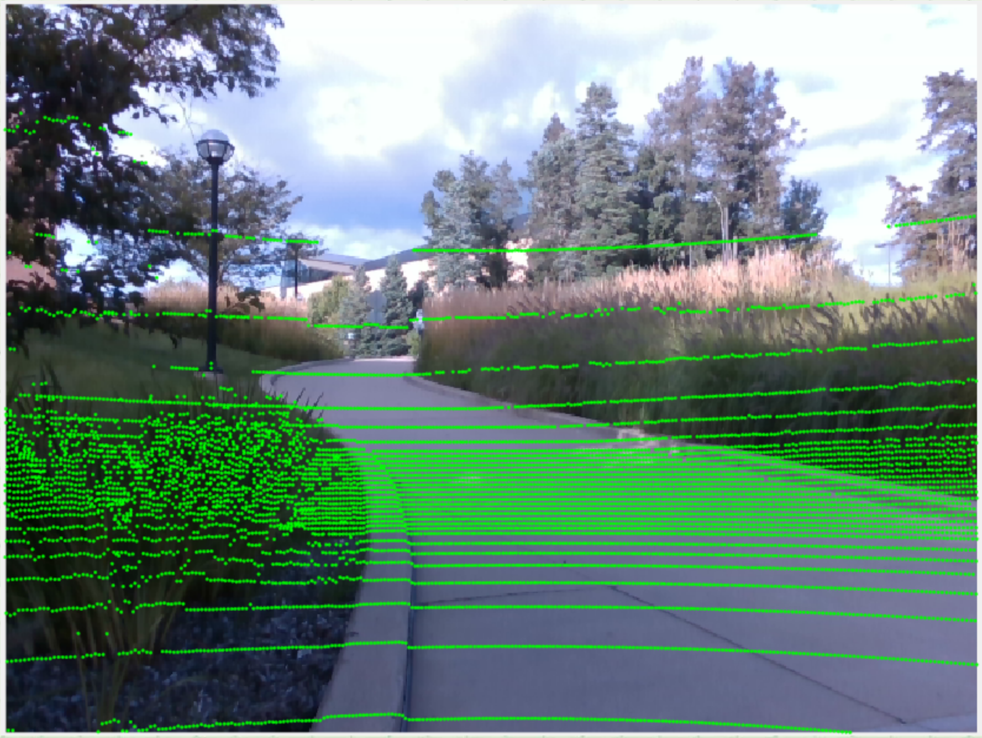
- Arg min should be thought of as another function in your toolbox,

$$\boxed{x^* = \underset{x \in \mathbb{R}^m}{\arg\min}\, f(x).}$$

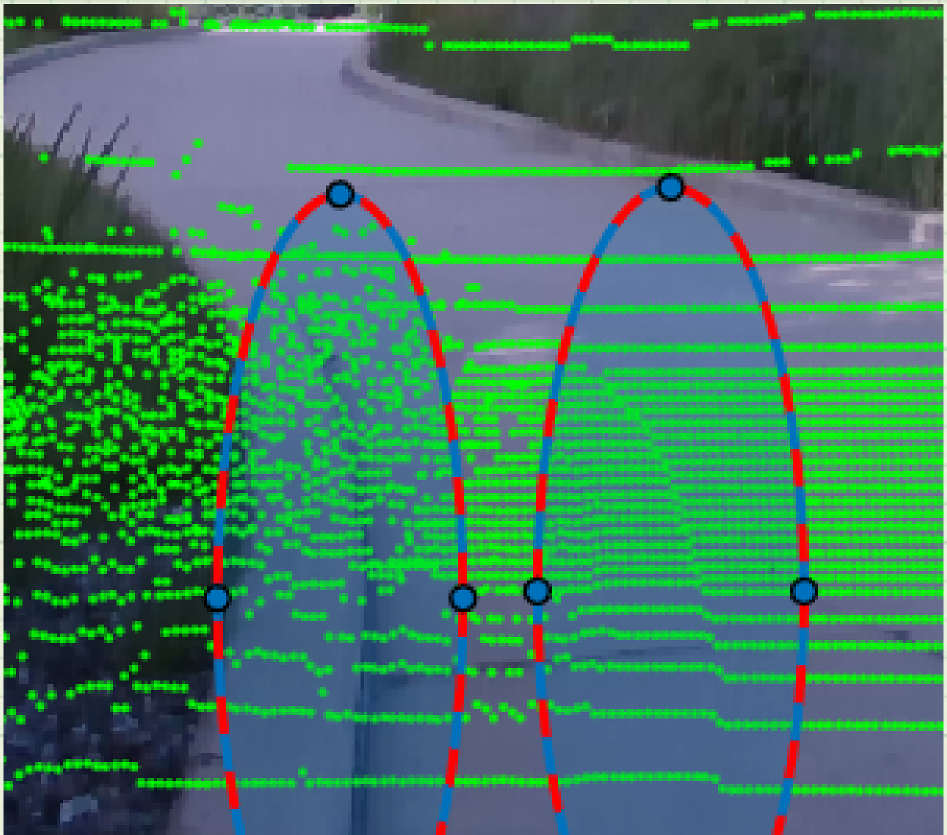
- Extrema of a function occur at places where the function's first derivative vanishes.
- The gradient of a function points in the direction of maximum rate of growth.
- We will add to our knowledge of derivatives, specifically, second derivative.
- Second-order optimization methods are based on root finding
- Convex functions have global minima
- Quadratic programs are special kinds of least squares problems



In this image, LiDAR and Camera are "calibrated" (means aligned)



Below, the curb is not aligned



Why optimization is important?

See above.

What is optimization?

Ingredients:

a) A set of potential solutions to a problem we care about.

For the LiDAR/Camera calibration problem, this would be the set of rotation matrices and translation vectors in $\mathbb{R}^3$.

We will assume our set of possible solutions is $\mathbb{R}^m$.

b) A way to distinguish between "good" solutions versus "bad" solutions.

$f: \mathbb{R}^m \longrightarrow \mathbb{R}$ cost function

The cost function allows us to compare two solutions and say which one we prefer.

a) least cost   minimization

b) most cost   maximization

In the branch of A.I. called Machine Learning, one use rather folksy and fun vocabulary

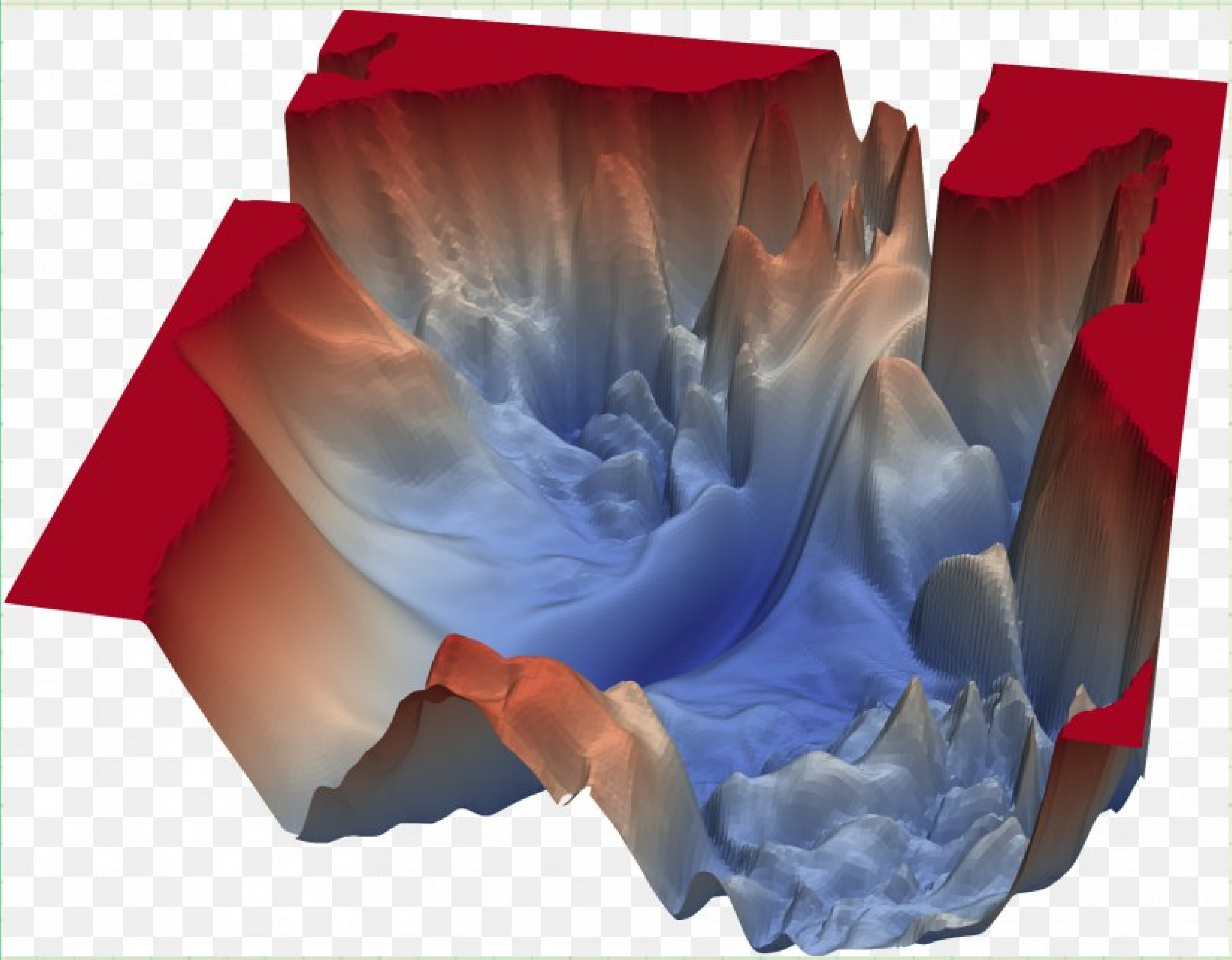
a) a cost $f: \mathbb{R}^m \to \mathbb{R}$ is called regret when on minimizes the function!

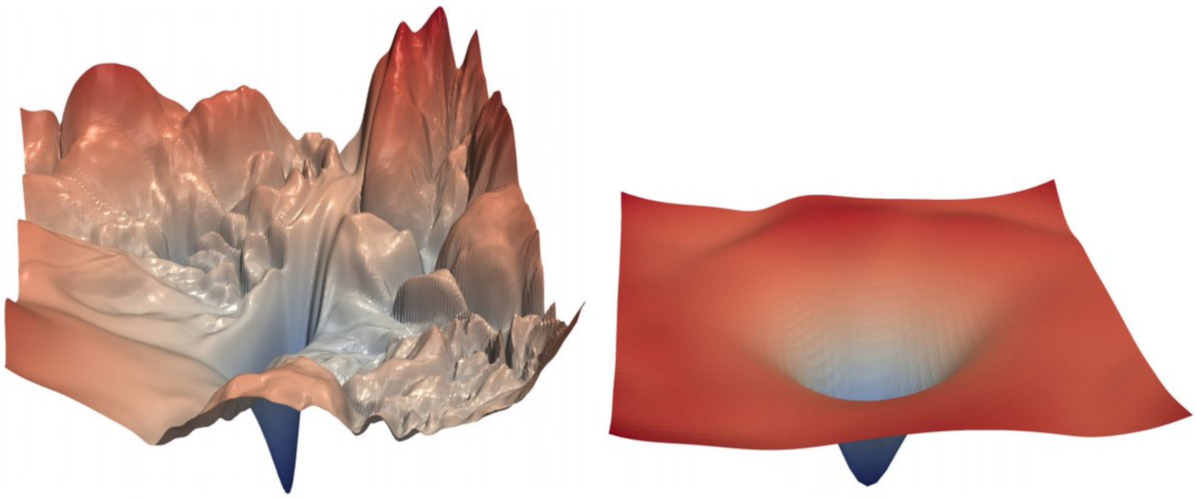
b) f is called a reward when doing maximization!

Designing cost functions for

a problem is absolutely the hardest part of the problem.

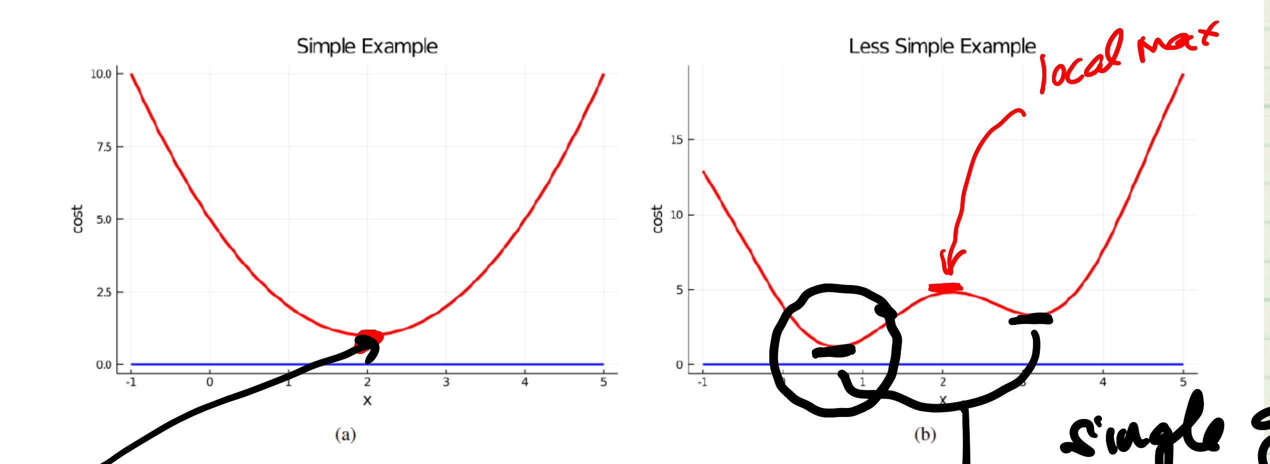
We will assume the cost has been given to us and work on algorithms to find minima of functions.





(a) without skip connections (b) with skip connections

Local vs Global minima



local max

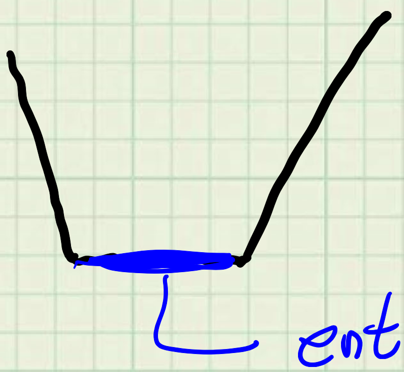
single global minimum

single global minimum and two local minima

Local Minimum = smallest value in a small region

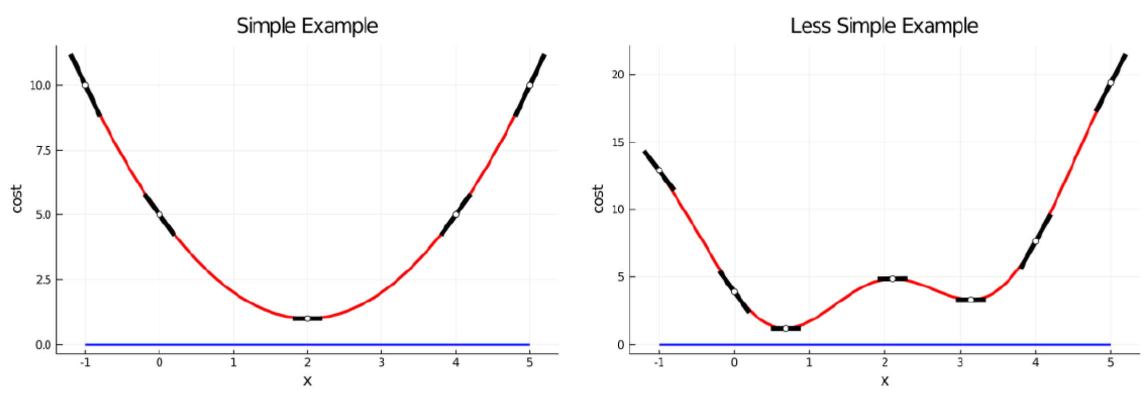
Our goal $x^* = \underset{x \in \mathbb{R}^m}{\arg\min}\, f(x)$

Abuse of notation !



cost(x)

entire set of minima



(a) (b)

Figure 2: The derivatives of the cost functions have been added at strategic points. (a) A "simple" cost function with a global minimum at $x^* = 2$ and (b) a "less simple" cost function where there are two local minima, one at $x^* \approx 0.68$ and one at $x^* \approx 3.14$, and a local maximum at 2.1.

Derivative = zero at local minima and maxima

Derivative indicates in which direction the function is increasing

$\frac{df(x_0)}{dx} > 0 \implies$ increasing to the right!

$\frac{df}{dx}(x_0) < 0 \implies$ increasing to the left!

Analytical view:

$$f(x) = f(x_k) + \frac{df}{dx}(x_k)(x - x_{k+1})$$

Seek x_{k+1} such that $f(x_{k+1}) < f(x_k)$

$$f(x_{k+1}) < f(x_k) + \frac{df}{dx}(x_k)(x_{k+1} - x_k)$$

We need $\frac{df}{dx}(x_k)(x_{k+1} - x_k) < 0$

$\frac{df}{dx}(x_k) > 0 \implies x_{k+1} - x_k < 0 \implies x_{k+1} < x_k$

$\frac{df}{dx}(x_k) < 0 \implies x_{k+1} - x_k > 0 \implies x_{k+1} > x_k$

$\frac{df}{dx}(x_k) \simeq 0$ we do not how to move in order to decrease $f(x_{k+1})$

Wed.

$$x_{k+1} = x_k - s \frac{df(x_k)}{dx} \quad , \quad s > 0 \text{ step size}$$