Numerical Optimization

Minimize a real-valued function f of n real variables,

$$\min_x f(x_0, x_1, \ldots, x_{n-1})$$

In vector notation: among $x \in \Re^n$, find $x^* = \operatorname{argmin} f(x)$

Usually we're more interested in x^* than in $p^* = \min f(x) = f(x^*)$

$f(x)$ is the "objective function" or (for DNNs) the "loss function"

Examples

- Best location (x_0, x_1) for a new cell tower
  - $f(x_0, x_1) =$ weakest signal in neighborhood
- Best cross-section for an airplane wing
  - $f(x_0, \ldots, x_{1000}) = -\text{lift}$ (to maximize lift)
- Least squares data fitting, $Ax \approx b$
  - $f(x) = \| Ax - b \|^2$
- Training deep neural nets: best weights $x_0, \ldots, x_{10^9}$
  - $f(x) =$ "loss function" $= \sum_{\text{training data}} (\text{DNN output} - \text{right answer})^2$

Possible additional features

- Sometimes there are *constraints* on x:
  - $Ax = b$, $Ax \leq b$, $x(i) \in \{0,1\}$, etc.
  - Convex constraints, nonlinear constraints, etc.

- Sometimes the function f has special features:
  - Convex
  - Linear
  - Smooth
  - Integer-valued

- In CS 111, we will only consider:
  - *Unconstrained* minimization (x can be anything in $\Re^n$)
  - *Convex* functions f (every chord is above the function values)
  - *Smooth* functions f (continuous, sometimes derivatives too)

Optimization: Scales and Algorithms

- $n = 1$ to 100 : Newton's method (dense)
- $n = 100$ to 10^4 : Newton (w/ sparse matrices)
- $n = 10^4$ to 10^6 : quasi-Newton, e.g. BFGS
- $n = 10^6$ to 10^8 : gradient descent (w/ acceleration)
- $n = 10^8$ to 10^{10+} : stochastic gradient descent (SGD)

(roughly speaking, with exceptions and caveats)