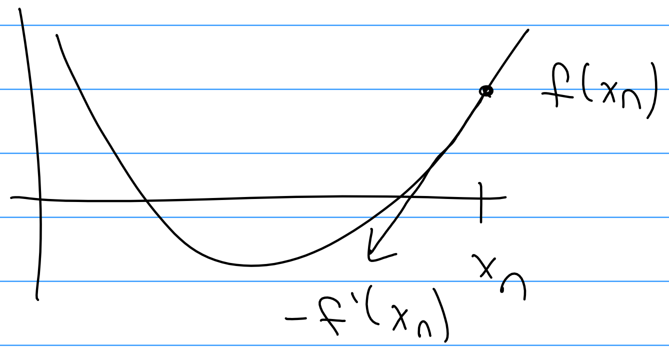
Last time: Minimization of single variable functions

Gradient Descent



$$x_{n+1} = x_n - \alpha_n f'(x_n)$$

α_n is chosen so that

$f(x_{n+1}) < f(x_n)$ → Line-search

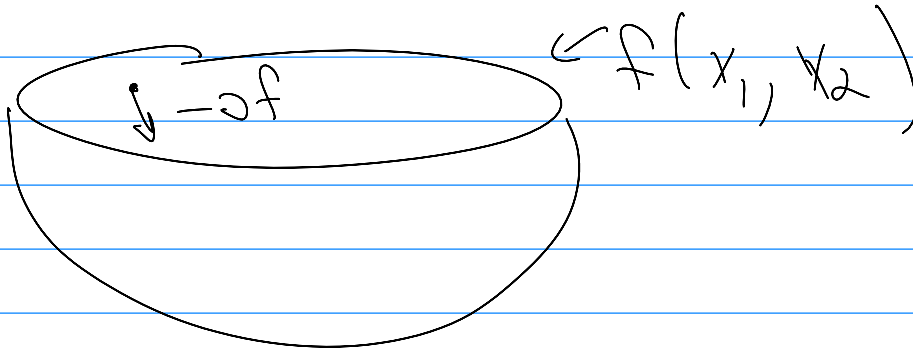
Multidimensional Minimization

① Gradient Descent

Extended to higher Dimensions:

$$\underline{x}_{n+1} = \underline{x}_n - \alpha_n \partial f(\underline{x}_n)$$

choose α_n such that $f(\underline{x}_{n+1}) < f(\underline{x}_n)$



You can show that if α_n satisfies the Wolfe Conditions (won't discuss) then the method will converge to a local minimum.

ex1) $$\alpha_n = \frac{(\underline{x}_n - \underline{x}_{n-1})^T (\nabla f(\underline{x}_n) - \nabla f(\underline{x}_{n-1}))}{\|\nabla f(\underline{x}_n) - \nabla f(\underline{x}_{n-1})\|_2^2}$$

Other methods might have faster convergence.

If $f(\underline{x})$ is convex, then the local minimum is the global minimum.

② Newton's method for minimization.

In multi dimensions:

$$f(\underline{x} + \underline{\Delta x}) = f(\underline{x}) + \underline{\Delta x}^T \nabla f(\underline{x}) + \frac{1}{2} \underline{\Delta x}^T \underline{H} \, \underline{\Delta x} + \text{H.O.T.}$$

$\underline{H}$ = Hessian = $\nabla\nabla f(\underline{x})$

$$= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} & \cdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \\ \vdots & & & \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

To find the minimum solve

$$\frac{\partial f}{\partial \underline{\Delta x}} = 0 \quad , \text{ ignoring } O(\underline{\Delta x}^2) \text{ terms \& higher}$$

$$\Rightarrow \underline{0} = \nabla f(\underline{x}_n) + \underline{H}(\underline{x}_n)\,\underline{\Delta x}$$

$$\Rightarrow \underline{\Delta x} = -\underline{H}^{-1}\nabla f$$

$$\underline{\Delta x} = \underline{x}_{n+1} - \underline{x}_n$$

$$\Rightarrow \underline{x}_{n+1} = \underline{x}_n - \underline{H}_n^{-1}\nabla f_n$$

Issues: What if $\underline{H}_n^{-1}$ does not exist or is ill-conditioned?

(3) Quasi-Newton Methods: Methods that use the concept behind Newton method but do <u>not</u> use the true Hessian.

Let $f(\underline{x})$ be the function to minimize, w/ $\underline{x}_n$ an approximation to the true minimum $\underline{x}^*$.

An <u>approximate</u> Taylor series is:

$$f(\underline{x}_n + \underline{\Delta x}) = f(\underline{x}_n) + \underline{\Delta x}^T \nabla f_n + \frac{1}{2} \underline{\Delta x}^T \underline{B}_n \underline{\Delta x}$$

Where $\underline{B} \approx \underline{H}$ that has nice properties (e.g., positive-definite)

Then define an iteration as

$$\underline{x}_{n+1} = \underline{x}_n - \alpha_n \underline{B}_n^{-1} \nabla f(\underline{x}_n)$$

w/ α_n chosen to minimize $f(\underline{x}_{n+1})$

The trick is how to compute $\underline{B}_n$.

General Method: Let $\underline{x}_0$ & $\underline{B}_0$ be given,

for $n=0$ to Convergence

$$\left.\begin{aligned} \underline{\Delta x}_n &= -\alpha_n \underline{B}_n^{-1} \nabla f(\underline{x}_n) \\ \underline{x}_{n+1} &= \underline{x}_n + \underline{\Delta x}_n \end{aligned}\right\} \text{line search}$$

$$\underline{y}_n = \nabla f(\underline{x}_{n+1}) - \nabla f(\underline{x}_n)$$

$$\underline{B}_{n+1} = \text{A function of } \underline{B}_n, \underline{\Delta x}_n, \underline{y}_n$$

A good choice for $\underline{B}_0$:

Let $\underline{\Delta x}_0 = -\alpha_0 \nabla f(\underline{x}_0)$, $\underline{y}_0 = \nabla f(\underline{x}_0 + \underline{\Delta x}_0) - \nabla f(\underline{x}_0)$

$$\underline{B}_0 = \frac{\underline{y}_0^T \underline{y}_0}{\underline{y}_0^T \underline{\Delta x}_0} \underline{I}$$

Different Methods have different Schemes to Compute $\underline{B}_{n+1}$

Davidon - Fletcher - Powell (DFP)

$$\underline{B}_{n+1} = \left(\underline{I} - \frac{\underline{y}_n \, \underline{\Delta x}_n^T}{\underline{y}_n^T \, \underline{\Delta x}_n} \right) \underline{B}_n \left(\underline{I} - \frac{\underline{\Delta x}_n \, \underline{y}_n^T}{\underline{y}_n^T \, \underline{\Delta x}_n} \right)$$

$$+ \frac{\underline{y}_n \, \underline{y}_n^T}{\underline{y}_n^T \, \underline{\Delta x}_n}$$

$$\underline{B}_{n+1}^{-1} = \underline{B}_n^{-1} + \frac{\underline{\Delta x}_n \, \underline{\Delta x}_n^T}{\underline{\Delta x}_n^T \, \underline{y}_n} - \frac{(\underline{B}_n^{-1}) \underline{y}_n \underline{y}_n^T (\underline{B}_n^{-1})}{\underline{y}_n^T (\underline{B}_n^{-1}) \underline{y}_n}$$

- Broyden - Fletcher - Goldfarb - Shannon (BFGS)

$$\underline{B}_{n+1}^{-1} = \left(\underline{I} - \frac{\underline{\Delta x}_n \, \underline{y}_n^T}{\underline{y}_n^T \, \underline{\Delta x}_n} \right) \underline{B}_n^{-1} \left(\underline{I} - \frac{\underline{y}_n \, \underline{\Delta x}_n^T}{\underline{y}_n^T \, \underline{\Delta x}_n} \right)$$

$$+ \frac{\underline{\Delta x}_n \, \underline{\Delta x}_n^T}{\underline{y}_n^T \, \underline{\Delta x}_n}$$

Others .....

Other minimization algorithms:

- Levenberg-Marquardt Method
- Nedler-Mead Simplex Method
- Trust-Region Methods

⋮

"Numerical Optimization" by
Nocedal & Wright