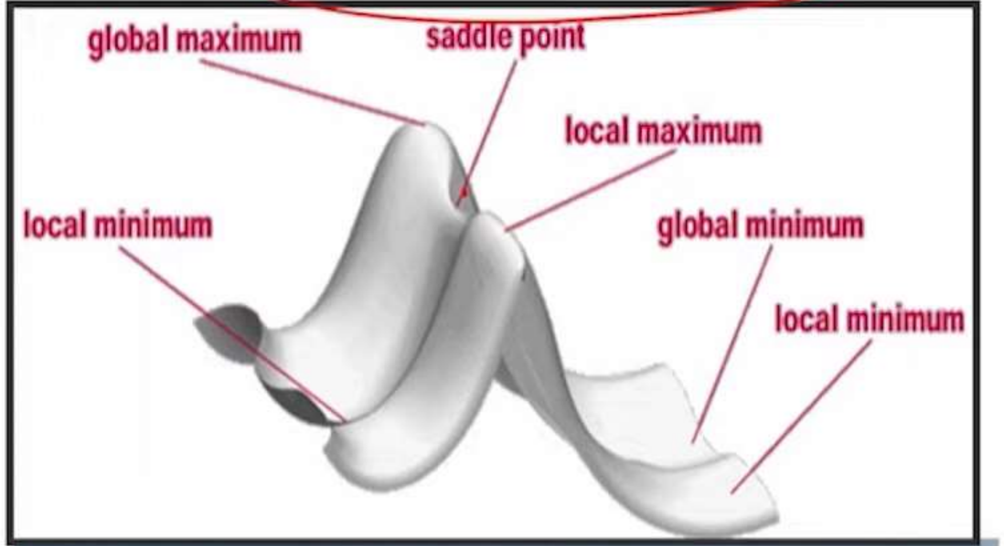
Unconstrained multivariate optimization - Directional search

- Aim is to reach the bottom most region
- Directions of descent
- Steepest descent
- Sometimes we might even want to climb the mountain for better prospects to get down further



Unconstrained multivariate optimization - Descent direction and movement

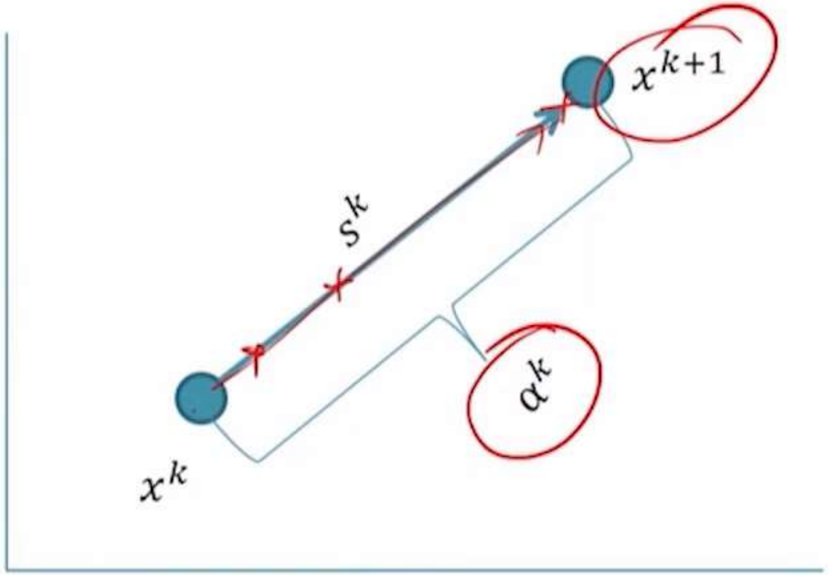
- Iterative

Scalar

$$x^{k+1} = x^k + \alpha^k s^k$$

Starting point | Step length | Search direction

$$n\times 1 \begin{pmatrix} s_1^k \\ s_2^k \\ \vdots \\ s_n^k \end{pmatrix} \qquad n\times 1 \begin{pmatrix} x_1^k \\ x_2^k \\ \vdots \\ x_n^k \end{pmatrix}$$



Multivariate Optimization
- Search direction computation
- Step length computation = Univariate optimization

- In ML techniques, this is called as the learning rule
- In neural networks
  - Back-propagation algorithm
  - Same gradient descent with application of chain rule
- In clustering
  - Minimization of an Euclidean distance norm

Steepest descent and optimum step size

- Minimize $f(x_1, x_2, \ldots x_n) = f(x)$
- **Steepest descent**
  - At iteration k starting point is x^k
  - Search direction s^k = Negative of gradient of f(x) $= -\nabla f(x^k)$
  - New point is $x^{k+1} = x^k + \alpha^k s^k$ where α^k is the value of α for which $f(x^{k+1}) = f(\alpha) =$ is a minimum (univariate minimization)

$x^0 \to x^*$

$f(x_1, x_2, \ldots, x_n)$

$s_1^k = -\nabla f = -\begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} = -\begin{pmatrix} g_1(x^k) \\ \vdots \\ g_n(x^k) \end{pmatrix}$

$\alpha = \alpha_{fixed}$

learning rate

$f(x^k + \alpha^k s^k)$

x^0

$x^1 = x^0 + \alpha^0 s^0$

$x^2 = x^1 + \alpha^1 s^1$

x^*