FYS-STK 4155, Sept 29, 2022

Taylor expansion

$C(\beta)$ $\qquad\qquad$ $\beta = \hat{\beta}$

$\beta - \beta^{(n)}$ $\qquad\qquad$ $g = \nabla_\beta C(\beta)$

$$C(\beta) = C(\beta^{(n)}) + g(\beta^{(n)})^T(\beta - \beta^{(n)}) + \frac{1}{2}(\beta - \beta^{(n)})^T H(\beta^{(n)})(\beta - \beta^{(n)}) + \ldots$$

$$b = \beta - \beta^{(n)}$$

$$C(\beta) \simeq C(\beta^{(n)}) + g^T b + \frac{1}{2} b^T H b \Rightarrow$$

$$\hat{\beta} = \beta^{(n)} - H^{-1} g(\beta^{(n)})$$

iterative expression

$$\beta^{(n+1)} = \beta^{(n)} - H^{-1}(\beta^{(n)})\, g(\beta^{(n)})$$

$$= \beta^{(n)} - \gamma^{(n)}\, g(\beta^{(n)})$$

learning rate

$$|\beta^{(n+1)} - \beta^{(n)}| \leq \varepsilon \sim 10^{-8} - 10^{-10}$$

$$C(\beta) \longrightarrow f(x) = \frac{1}{2} x^T A x - x^T b$$

$$\frac{\partial f(x)}{\partial x} = 0 = Ax - b$$

Linear algebra

$$Ax = b$$

<u>Steepest Descent</u>

Define a residual error

(R) $r = b - Ax$

when $r=0$, have solution

we start with a guess for $x \simeq x_0$

$$r_0 = b - Ax_0 \quad (x_0 = 0)$$

After $k+1$ iterations

$$r_{k+1} = b - Ax_{k+1}$$

<u>assume</u>

$$x_{k+1} = x_k + \alpha_k r_k$$

$$r_{k+1} = b - A(x_k + \alpha r_k)$$

$$= \underbrace{(b - Ax_k)}_{r_k} - \alpha_k A r_k$$

$$r_{k+1}^T r_k = 0$$

Multiply with r_k^T

$$0 = r_k^T r_k - \alpha_k r_k^T A r_k \Rightarrow$$

$$\alpha_k = \frac{r_k^T r_k}{r_k^T A r_k}$$

$$x_{k+1} = x_k + \alpha_k r_k$$

$$\frac{\partial f(x)}{\partial x} = Ax - b = -\underbrace{(b - Ax)}_{r_k}$$

$$\nabla_x f(x) = g(x) \Rightarrow$$

$$x_{k+1} = x_k - \alpha_k g(x)$$

$$\frac{r_k^T r_k}{r_k^T A r_k}$$

$$= \boxed{\frac{g_k^T g_k}{g_k^T A g_k}}$$ expensive

$$x_{k+1} = x_k - \gamma g(x)$$

Example

$$f(x_1\, x_2) = f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right)$$

$$= \frac{1}{2} \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$- \begin{bmatrix} 5 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= x_1^2 + x_1 x_2 + 10 x_2^2 - 5x_1 - 3x_2$$

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 2x_1 + x_2 - 5 = 0$$

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = x_1 + 20x_2 - 3 = 0$$

$$\frac{\partial f}{\partial x} = Ax - b$$

$$f(x) = \frac{1}{2} x^T A x - x^T b$$

iterative scheme

$$x_{i+1} = x_i - \gamma_i \nabla_x f(x_i)$$

For a suitable step size

$$f(x_0) \geq f(x_1) \geq f(x_2) \ldots$$

till it reaches a local minimum.

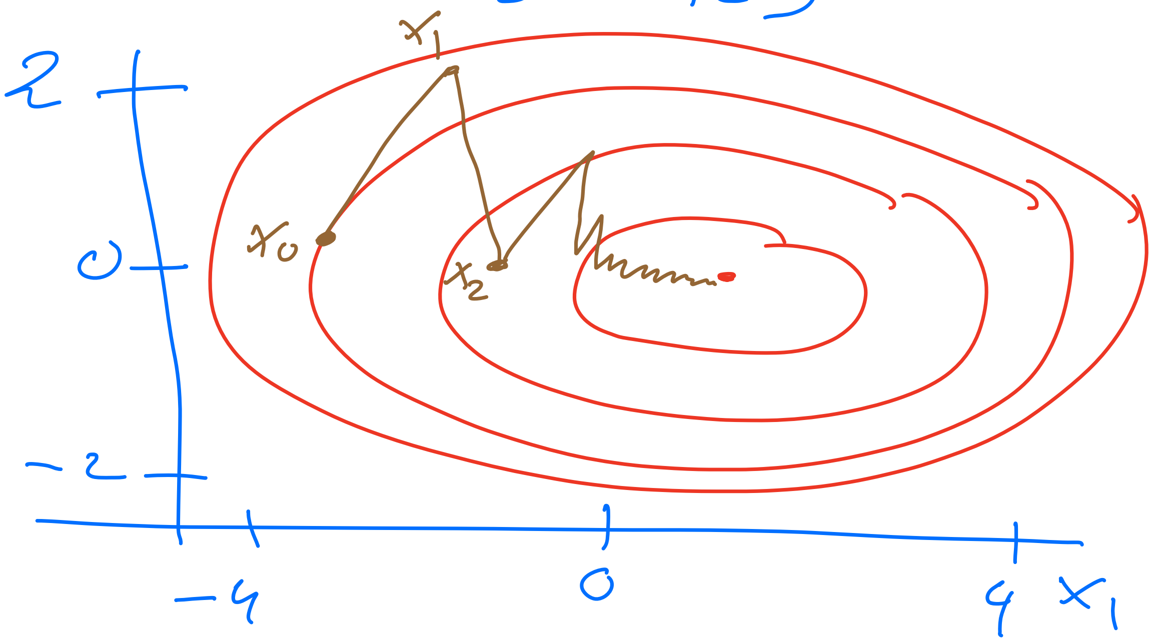
$$\gamma = 0.085$$

guess $x_0 = \begin{bmatrix} -3 \\ 1 \end{bmatrix}$

$$x_1 = x_0 - \gamma_0 \cdot \nabla_x f(x_0)$$

$$x_1 = \begin{bmatrix} -1.98 \\ 1.21 \end{bmatrix}$$

$$x_2 = \begin{bmatrix} -1.32 \\ -0.42 \end{bmatrix}$$



Overarching themes

- efficient of $g(x)$
- ways to update the learning rate & without

computing for example
the inverse of H
or for steepest descent

$$\gamma_k = \frac{g^T g}{g^T H g}$$

Gradient descent with
momentum (memory)

Motivation: Newton's
equation of motion with
external force and a
drag/friction

$$F(x) = -\nabla_x V(x)$$

$$m \frac{d^2 x}{dt^2} + \mu \frac{dx}{dt} = -\nabla_x V(x)$$
$$= F(x)$$

$$\frac{d^2 x}{dt^2} \simeq \frac{x_{t+\Delta t} - 2x_t + x_{t-\Delta t}}{(\Delta t)^2}$$

$$\frac{dx}{dt} \simeq \frac{x_{t+\Delta t} - x_t}{\Delta t}$$

Define

$$\Delta x_{t+\Delta t} = x_{t+\Delta t} - x_t$$

$$\Delta x_t = x_t - x_{t-\Delta t}$$

$$\Delta x_{t+\Delta t} = -\frac{(\Delta t)^2}{m+\mu\Delta t} \nabla V(x)$$

$$+ \frac{m}{m+\mu\Delta t} \Delta x_t$$

Define $\quad \delta = \dfrac{m}{m+\mu\Delta t}$

$$\gamma = \frac{(\Delta t)^2}{m+\mu\Delta t}$$

$$\Delta x_{t+\Delta t} = -\gamma \nabla_x V(x) + \delta \Delta x_t$$

memory parameter

$$\delta = \frac{m}{m + \mu \Delta t} \qquad \delta \in [0, 1]$$

when μ large

$$\delta \simeq \frac{m}{\mu \Delta t}$$

$X_t \to \beta_i \qquad x_{t+\Delta t} \to \beta_{i+1}$

$\nabla_x V(x) \Rightarrow g(\beta_i)$

$$\Delta \beta_{i+1} = \beta_{i+1} - \beta_i$$

$$\Delta \beta_i = \beta_i - \beta_{i-1} \quad \Rightarrow$$

$$\beta_{i+1} = \beta_i - \gamma_i g(\beta_i) + \delta(\beta_i - \beta_{i-1})$$

often rewritten as

$$v_i = \delta(\beta_i - \beta_{i-1}) - \gamma_i g(\beta_i)$$

$$\beta_{i+1} = \beta_i + v_i$$

Algorithm

- require learning rate γ

- require memory parameter
  $\delta \in [0, 1]$

initialize β_0 and v_0

while stopping criterion not met

- compute gradient
- compute v_i
- apply v_i to update
  $\beta_{i+1} = \beta_i + v_i$

end while