Hacking Steepest Descent for Better Convergence

Extrapolation methods: "momentum"

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) + \beta_k (x_k - x_{k-1})$$

Heavy ball method:

Gradient descent: $\frac{\kappa(A)-1}{\kappa(A)+1}$.

$$\alpha_k = \alpha, \quad \beta_k = \beta$$

$$\|e_{k+1}\|_A = \frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\|e_k\|_A$$

Demo: Steepest Descent [cleared] (Part 2)

Optimization in Machine Learning

What is *stochastic gradient descent* (*SGD*)?

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x).$$

f_i : "data point" / observation.

GD : $x_{k+1} = x_k - \alpha \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x).$

SGD : $x_{k+1} = x_k - \alpha \nabla \underline{\underline{f_{\phi(k)}}}(x_k)$

"minibatch".

"ADAM"

Conjugate Gradient Methods

Can we optimize in *the space spanned* by the last two step directions?

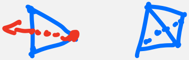
$$x^T A y = 0 \quad \text{"conjugate"}$$

$$(\alpha_k, \beta_k) = \operatorname{argmin}\left[f(x_k - \alpha_k \nabla f(x_k) + \beta_k (x_k - x_{k-1}) \right]$$

Demo: Conjugate Gradient Method [cleared]

Nelder-Mead Method

Idea:



= "simplex"

Demo: Nelder-Mead Method [cleared]

Newton's method (n D)

What does Newton's method look like in n dimensions?

① $\nabla f(x) = 0$.

② $f(x+s) \approx f(x) + \nabla f(x)^T s + \frac{1}{2} s^T H_f(x) s = \hat{f}(s)$

$\nabla \hat{f}(s) = 0 \Rightarrow H_f(x) s = -\nabla f(x)$.

step 1. x_0.

" 2. solve $\underline{H_f(x) s_k = -\nabla f(x_k)}$ for s_k

" 3. $x_{k+1} = x_k + s_k$

Newton's method (n D): Observations

Drawbacks?

1. need for 2nd derivatives.
2. local convergence.
3. $H_f(x)$ can be close to indefinite.

Demo: Newton's method in n dimensions [cleared]

Quasi-Newton Methods

Secant/Broyden-type ideas carry over to optimization. How?

- Come up with approximation to Hessian.
- Secant cond.

BFGS: Secant-type method, similar to Broyden:

$$\boldsymbol{B}_{k+1} = \boldsymbol{B}_k + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k} - \frac{(\boldsymbol{B}_k \boldsymbol{s}_k)(\boldsymbol{s}_k^T \boldsymbol{B}_k)}{\boldsymbol{s}_k^T \boldsymbol{B}_k \boldsymbol{s}_k}$$

where

- $\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$
- $\boldsymbol{y}_k = \nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k)$

$B_k s_k = y_k$.

In-Class Activity: Optimization Methods

In-class activity: Optimization Methods

Nonlinear Least Squares: Setup

What if the f to be minimized is actually a 2-norm?

$$f(\boldsymbol{x}) = \|\boldsymbol{r}(\boldsymbol{x})\|_2, \qquad \boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{y} - \boldsymbol{a}(\boldsymbol{x})$$

linear : $\|\underbrace{Ax-b}\|_2$ $\quad \underbrace{A^TAx = A^Tb}$

nonliner : $\|r(x)\|_2$

$$\varphi(x) := \frac{1}{2} r(x)^T r(x) = \frac{1}{2} f^2(x)$$

minimize $\varphi(x)$ instead.

$$\frac{\partial}{\partial x_i}\varphi = \frac{1}{2}\sum_{j=1}^{n} \underbrace{\frac{\partial}{\partial x_i}\left[r_j(x)^2\right]} = \sum_{j=1}^{n} \frac{\partial r_j}{\partial x_i} \cdot r_j$$

$$\nabla \varphi = J_r(x)^T r(x)$$

Gauss-Newton

For brevity: $J := J_{\boldsymbol{r}}(\boldsymbol{x})$.

$$H\varphi(x) = J^T J + \sum_i r_i H_{r_i}(x)$$

Newton : $H\varphi(x)\, s = -\nabla\varphi$.

Gauss-Newton : $J^T J s = -\nabla\varphi = -J^T r(x)$.

$$J s \cong -r(x)$$

linear least square!