Recitation #13

Irina Espejo (iem244@nyu.edu)

Center for Data Science

DS-GA 1014: Optimization and Computational Linear Algebra for Data Science

recall

Gradient Descent

Let $f : \mathbb{R}^n \to \mathbb{R}$. Gradient descent is an algorithm to find minimize a convex and twice differentiable function. The update algorithm for gradient descent is:

$$x_{t+1} = x_t - \alpha_t \nabla f(x_n)$$

for $\alpha_t \in \mathbb{R}$ is the step size at time t

Convergence

Convergence is ensured if the function is convex and twice differentiable. The speed of convergence follows the inequality:

$$f(x_t) - f(x^\star) \leq \frac{2L\|x_0 - x^\star\|^2}{t+4}$$

Alert! Many times in Data Science we have non-convex functions, generally we still apply methods derived from gradient descent but convergence is not ensured.

recall

Newton's method

This is basically adjusting the step α_t to the "optimal step" under "nice" conditions of f:

$$x_{t+1} = x_t - H f(x_t)^{-1} \nabla f(x_t)$$

If we take the step $\alpha_t = F f(x_t)$ and the function f is nice then convergence is super super fast $\|x_t - x^\star\|^2 \le Ce^{-a2^t}$

L-smooth
μ - strong

Taylor expansion at second order

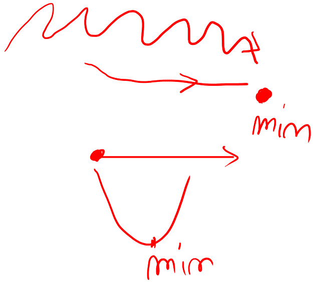
Accelerating the descent

gradient descent with momentum

The upgrade algorithm for an accelerated gradient descent is

$$x_{t+1} = x_t + v_t \quad \text{momentum}$$

where the velocity $v_t = \overbrace{-\alpha_t \nabla f(x_t)}^{\text{GD}} + \beta_t v_{t-1}$ adds momentum to the descent trajectory to get to the minimum faster (under "nice" conditions of the function f)



Exercise 1 (ex 0.14 2019 review)

Assume we use gradient descent when minimizing the least-square cost $f(x) = \|Ax - y\|^2$. Write the gradient descent step update for this problem. //

$x \in \mathbb{R}^n \quad A \in \mathbb{R}^{m \times n}$

$\boxed{\nabla f = 0} \rightarrow x^*$

$x^* = \min_{x \in \mathbb{R}^n} f(x)$

$f(x) = \|Ax - y\|^2$

$f(x) = \langle Ax - y, Ax - y \rangle$

GD update rule: $x_{t+1} = x_t - \alpha_t \nabla f(x_t)$

$\nabla f(x_t) = 2(Ax_t - y)A^T$

$$\boxed{x_{t+1} = x_t - \alpha_t 2(Ax_t - y)A^T}$$

Exercise 1

practice: gradient descent

Exercise 1 (ex 0.14 2019 review)

If now we use a Newton update, which is the gradient descent update?

$$\min_{x \in \mathbb{R}^n} f(x) \qquad f(x) = \|Ax - y\|^2$$

$$\nabla f(x_t) = 2(Ax_t - y)A^T$$

$$H = 2AA^T$$

$$x_{t+1} = x_t - \frac{1}{\cancel{2}}(AA^T)^{-1}\cancel{2}(Ax_t - y)A^T$$

Exercise 1

$$= x_t - (AA^T)^{-1} A x_t A^T$$

$$+ (AA^T)^{-1} y A^T$$

$$A^T x$$

$$x^T A$$

$$= x_t - (A^T)^{-1} \cancel{A^{-1}} \cancel{A} x_t A^T + (A^T)^{-1} y A^T$$

practice: gradient descent

Exercise 1 (ex 0.14 2019 review)

What happens to the speed of gradient descent for linear regression when we first perform some dimensionality reduction on the features?

$$\| f(x_t) - f(x^*) \| \xrightarrow[\text{how fast?}]{} 0$$

$$f(x) = \| Ax - y \|^2 \quad x \in \mathbb{R}^n \quad x = \begin{pmatrix} - \\ - \\ - \\ - \\ - \end{pmatrix}$$

$$x \longrightarrow x' = \begin{pmatrix} - \\ - \end{pmatrix} \in \mathbb{R}^{n'} \quad \text{Now, } \| f(x_t') - f(x^*) \| \;??$$

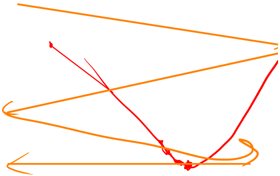
$$n' \ll n$$

Exercise 1

$X = \begin{pmatrix} \\ \\ \end{pmatrix} \rightarrow x = (\quad)$

$\nabla f = (\quad)$

$H = (\quad)$



$X_{t+1} = X_t - \alpha_t \nabla f(X_t)$

$X_{t+1} = X_t - (H)^{-1} \nabla f(X_t)$ Keep ↑ λ of H

remap ↓ λ of H

I

Exercise 1 (ex 0.14 2019 review)

Think about possible stopping criteria for the gradient descent algorithm.

$$x_{t+1} = x_t - \alpha_t \nabla f(x_t) \qquad t = 0 \dots N$$

π

t	x_t
0	0'1227
1	0'10
2	0'8
3	
⋮	
100	0'05001
100	0'050011

$x^* = 0'050$

Stochastic Gradient Descent

we use this variant of gradient descent when the function to optimize is stochastic. Plus, instead of calculating the full gradient we calculate a cheaper but noisy gradient. Turns out under suitable conditions, this algorithm will converge to a local minimum.

$X_1 \ldots X_n$ $\quad \mathbb{E}_{p(X)}[f(X)]$

SGD $X_1, \ldots, X_n \rightarrow$ take one X_i

$X_{t-1} = X_t - \alpha_t \nabla_i f(X)$

review final exam