Optimization

They see me rollin'

Sahit Chintalapudi

April 4, 2018

Overview

1. "Classical" Optimization
   - Levenberg-Marquadt
2. Modern/DL Approaches to Optimization
   - Stochastic Gradient Descent
   - Natural Gradients
   - Adam

Gauss Newton - Setup

- we want to minimize the function: $r : \mathbb{R}^n \to \mathbb{R}^m$

$$||r(x)||^2 = \sum_{i=1}^{n} r_i(x)^2$$

- Linearize with a (first-order) Taylor Expansion. Let $J(x_k)$ be the Jacobian of r at x_k
- x_0 acts as our initial guess of where the minimum is

$$r(x_k) \approx r(x_k) + J(x_k)(x - x_k)$$

Gauss Newton - Nonlinear Least Squares

- If we let $A_k = J(x_k)$ and $b_k = J(x_k)x_k - r(x_k)$

$$||r(x)||^2 \approx ||r(x_k) + J(x_k)(x - x_k)||^2 = ||A_k x - b_k||^2$$

- (recall) Solved by

$$x_{k+1} = (A_k^T A_k)^{-1} A_k b_k$$

- no guarantee of convergence!

Gradient Descent

- Move in the direction of the gradient

$$x_{k+1} = x_k - \eta J(x_k)$$

- Always converges (but sometimes to local optima)

Levenberg-Marquadt

- A combination of gradient descent and Gauss Newton
- Take steps in gradient descent direction at first
- Approaches Gauss-Newton with smaller steps and accelerates to a minimum

$$x_{k+1} = x_k - (J^T J + \lambda I)^{-1} J(x_k)$$

Stochastic Gradient Descent

- Approximate true cost function by measuring error on individual update term
- Adding a momentum term gives a nifty new update rule

$$x_{k+1} = x_k - (\eta J(x_k) + \alpha \Delta x)$$

- Better computational performance

Natural Gradient Descent - background

Definitions: (Sorry)
Kullback-Leibler (KL) Divergence: A metric for measuring how two distributions diverge

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

Between two normal distributions:

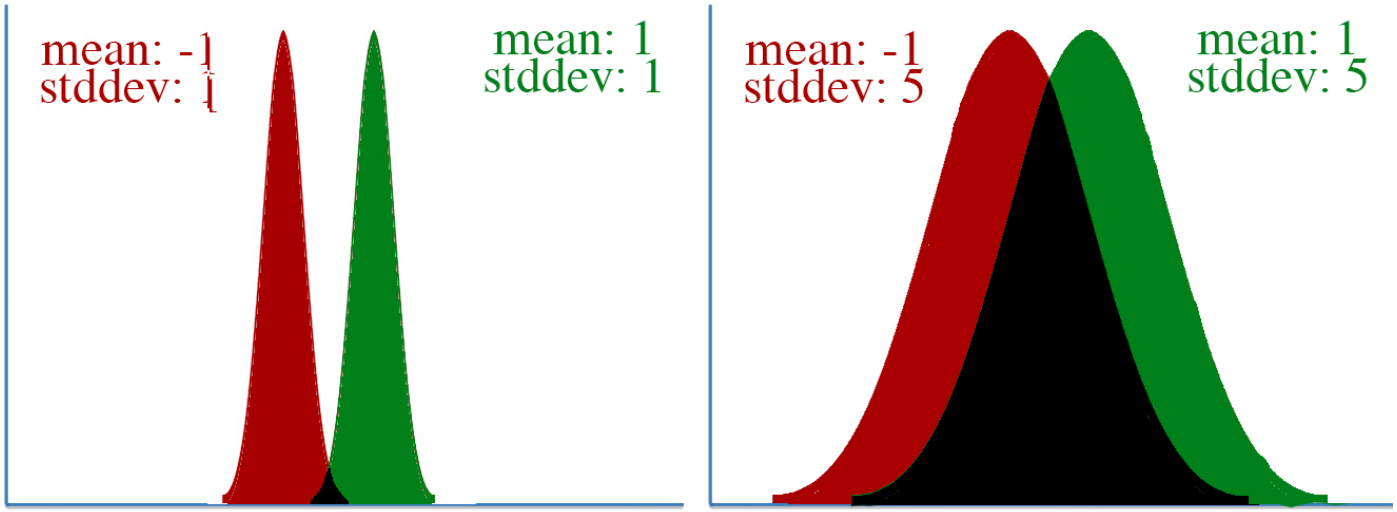
$$KL(p, q) = \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

Fisher Information Matrix: Gives us the "curvature" of the KL-divergence

$$|\mathcal{I}(\theta)_{ij}| = (\frac{\partial^2}{\partial\theta_i'\theta_j'} D(\theta||\theta'))$$

Natural Gradient Descent - Idea

- Standard gradient descent allows shift in the weight distributions that can be larger in the context of the weights as a whole



- We can take steps inversely proportional to the fisher information matrix to control the change in parameter distrubution

$$\text{naturalGrad} = \text{inverse}(\text{fisher}) * \text{sgd}$$

Adagrad

- What if we had different learning rates for different weights?

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot g_t$$

- G_t is a diagonal matrix containing sum of squared gradients
- Don't have to tune a learning rate!
- If one wishes to obtain something, something of equal value must be given. This is the law of equivalent exchange

Adaptive Moment Estimation (Adam)

- Store a decaying average of past squared gradients and past gradients

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

- This (along with some bias corrections...) tgives us the Adam update rule

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

Theorem 4.1. *Assume that the function f_t has bounded gradients, $\|\nabla f_t(\theta)\|_2 \leq G$, $\|\nabla f_t(\theta)\|_\infty \leq G_\infty$ for all $\theta \in R^d$ and distance between any θ_t generated by Adam is bounded, $\|\theta_n - \theta_m\|_2 \leq D$, $\|\theta_m - \theta_n\|_\infty \leq D_\infty$ for any $m, n \in \{1, ..., T\}$, and β_1, $\beta_2 \in [0, 1)$ satisfy $\frac{\beta_1^2}{\sqrt{\beta_2}} < 1$. Let $\alpha_t = \frac{\alpha}{\sqrt{t}}$ and $\beta_{1,t} = \beta_1 \lambda^{t-1}$, $\lambda \in (0, 1)$. Adam achieves the following guarantee, for all $T \geq 1$.*

$$R(T) \leq \frac{D^2}{2\alpha(1-\beta_1)} \sum_{i=1}^{d} \sqrt{T\widehat{v}_{T,i}} + \frac{\alpha(1+\beta_1)G_\infty}{(1-\beta_1)\sqrt{1-\beta_2}(1-\gamma)^2} \sum_{i=1}^{d} \|g_{1:T,i}\|_2 + \sum_{i=1}^{d} \frac{D_\infty^2 G_\infty \sqrt{1-\beta_2}}{2\alpha(1-\beta_1)(1-\lambda)^2}$$

Our Theorem 4.1 implies when the data features are sparse and bounded gradients, the summation term can be much smaller than its upper bound $\sum_{i=1}^{d} \|g_{1:T,i}\|_2 << dG_\infty\sqrt{T}$ and $\sum_{i=1}^{d} \sqrt{T\widehat{v}_{T,i}} << dG_\infty\sqrt{T}$, in particular if the class of function and data features are in the form of section 1.2 in (Duchi et al., 2011). Their results for the expected value $\mathbb{E}[\sum_{i=1}^{d} \|g_{1:T,i}\|_2]$ also apply to Adam. In particular, the adaptive method, such as Adam and Adagrad, can achieve $O(\log d\sqrt{T})$, an improvement over $O(\sqrt{dT})$ for the non-adaptive method. Decaying $\beta_{1,t}$ towards zero is important in our theoretical analysis and also matches previous empirical findings, e.g. (Sutskever et al., 2013) suggests reducing the momentum coefficient in the end of training can improve convergence.

Finally, we can show the average regret of Adam converges,

Corollary 4.2. *Assume that the function f_t has bounded gradients, $\|\nabla f_t(\theta)\|_2 \leq G$, $\|\nabla f_t(\theta)\|_\infty \leq G_\infty$ for all $\theta \in R^d$ and distance between any θ_t generated by Adam is bounded, $\|\theta_n - \theta_m\|_2 \leq D$, $\|\theta_m - \theta_n\|_\infty \leq D_\infty$ for any $m, n \in \{1, ..., T\}$. Adam achieves the following guarantee, for all $T \geq 1$.*

$$\frac{R(T)}{T} = O(\frac{1}{\sqrt{T}})$$

This result can be obtained by using Theorem 4.1 and $\sum_{i=1}^{d} \|g_{1:T,i}\|_2 \leq dG_\infty\sqrt{T}$. Thus, $\lim_{T\to\infty} \frac{R(T)}{T} = 0$.

ty

- https://see.stanford.edu/materials/lsoeldsee263/07-ls-reg.pdf
- http://people.duke.edu/ hpgavin/ce281/lm.pdf
- http://kvfrans.com/what-is-the-natural-gradient-and-where-does-it-appear-in-trust-region-policy-optimization/
- http://ruder.io/optimizing-gradient-descent/
- Rapha et Manas