ECEn 671: Mathematics of Signals and Systems
Moon: Chapter 14.

Randal W. Beard

Brigham Young University

August 29, 2023

Table of Contents

Section 1

Gradient Descent

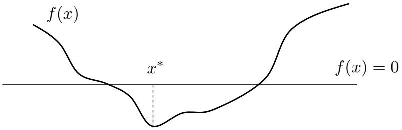
Gradient Descent

The topic for the remainder of the course is minimization and maximization of functions.

In particular we will constrain our attention to continuously differentiable functions.

Gradient Descent

Suppose we have a function of the form



and we would like to find x^*, what should we do?

Gradient Descent

The basic idea of gradient descent is to pick any $x^{[0]}$ and then move "downward". To move down, we look at the slope of f.

If $\frac{\partial f}{\partial x}(x^{[k]})$ is positive, chose $x^{[k+1]} < x^{[k]}$.

If $\frac{\partial f}{\partial x}(x^{[k]})$ is negative, choose $x^{[k+1]} > x^{[k]}$

i.e.

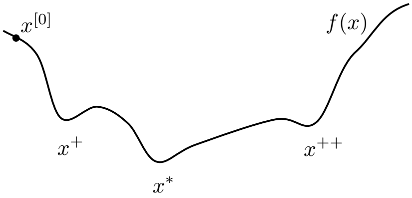
$$x^{[k+1]} = x^{[k]} - \alpha \frac{\partial f}{\partial x}(x^{[k]}),$$

where α is the step size.

Gradient Descent

Before moving to the multivariable case, lets consider the potential problems with this approach.

Problem 1: Local Minima. If f looks like this:



then if the initial condition is at $x^{[0]}$, the iteration

$$x^{[k+1]} = x^{[k]} - \alpha \frac{\partial f}{\partial x}(x^{[k]})$$

will converge to x^{+}, if α is small enough.

Gradient Descent

Other initial conditions will result in x^{++} while others will give x^*, the true minimum.

This is a fundamental problem with <u>any</u> method that relies on derivative information. There are no completely satisfactory solutions to the problem. However there are many ad-hoc fixes.

Example

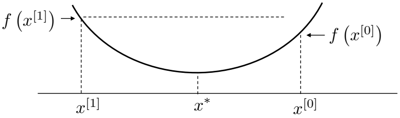
- Execute from numerous "random" initial conditions and pick the lowest solution.
- Occassionally introduce random jumps in x.
- etc...

Gradient Descent

Problem 2: Step Size. The selection of α can have a major effect on the convergence of the sequence

$$x^{[k+1]} = x^{[k]} - \alpha \frac{\partial f}{\partial x}(x^{[k]})$$

For example,



Note f is very steep on sides, so $\alpha \frac{\partial f}{\partial x}(x^{[k]})$ could be large. This could cause $x^{[1]}$ to overshoot the minimum. This could cause (1) instability, (2) limit cycles, (3) extremely slow and oscillatory convergence

Gradient Descent

Lesson: Don't make α too large.

However if α is too small, then convergence will be very slow.

Most implementations adapt the size of α.

Section 2

Gradient Descent: Multivariable Case

Gradient Descent: Multivariable Case

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a multivariable function.

Example

If $x \in \mathbb{R}^n$ then $f(x) = x_1^2 + x_2^2 + \cdots + x_n^2$ maps $\mathbb{R}^n \to \mathbb{R}$.

The gradient of a multivariable function is

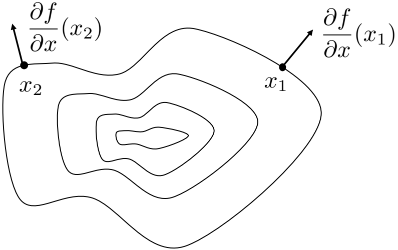
$$\frac{\partial f}{\partial x} = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

and maps $\mathbb{R}^n \to \mathbb{R}^n$.

Gradient Descent: Multivariable Case

Example

If $f(x) = x_1^2 + \cdots + x_n^2$ then $\frac{\partial f}{\partial x} = \begin{pmatrix} 2x_1 \\ 2x_2 \\ \vdots \\ 2x_n \end{pmatrix}$



The gradient points perpendicular to the level curves of f.

Gradient Descent: Multivariable Case

Theorem (Moon Theorem 14.5)

Let $f : \mathbb{R}^m \to \mathbb{R}$ *be a differentiable function in some open set* D*. The gradient* $\frac{\partial f}{\partial x}(x)$ *points in the direction of the maximum increase of* f *at the point* x*.*

Gradient Descent: Multivariable Case

Proof.

Expand $f(x + \lambda y)$ in a Taylor series as

$$f(x + \lambda y) = f(x) + \lambda \frac{\partial f}{\partial x}^T (x) y + \text{Higher Order Terms (HOT)}$$

where HOT. are $O(\lambda^2)$, i.e.,

$$\lim_{\lambda \to 0} \frac{H.O.T.}{\lambda} = 0.$$

We would like to find y that maximizes $f(x + \lambda y)$ as $\lambda \to 0$.
By Cauchy-Schwartz, $\frac{\partial f}{\partial x}^T y$ is maximized when $y = \frac{\partial f}{\partial x}$. $\square$

Gradient Descent: Multivariable Case

For multivarible functions, the gradient descent formula is

$$\boxed{x^{[k+1]} = x^{[k]} - \alpha_k \frac{\partial f}{\partial x}(x^{[k]})}$$

Again, the selection of the step size is very important. If α_k is too small convergence will be slow.

If α_k is too large, algorithm could be unstable.

How to pick the right α?

Gradient Descent: Multivariable Case

Locally around a min or max, every smooth function can be approximated by a quadratic (Taylor series).

We can gain insight about the selection of α by studying quadratic functions.

Let $f(x) = x^T Rx - 2b^T x$ where $x \in \mathbb{R}^m, b \in \mathbb{R}^m, R = R^T > 0$.

Taking the gradient we get

$$\frac{\partial f}{\partial x} = 2Rx - 2b.$$

Gradient Descent: Multivariable Case

So the gradient descent algorithm is

$$x^{[k+1]} = x^{[k]} - 2\alpha(Rx^{[k]} - b).$$

Let x^* satisfy $Rx^* = b$ then

$$x^{[k+1]} - x^* = x^{[k]} - x^* - 2\alpha(Rx^{[k]} - Rx^*)$$

Define $y^{[k]} = x^{[k]} - x^*$ and $\mu = 2\alpha$, then

$$\begin{aligned} y^{[k+1]} &= y^{[k]} - \mu R y^{[k]} \\ &= (I - \mu R) y^{[k]} \\ \implies y^{[k]} &= (I - \mu R)^k y^{[0]}. \end{aligned}$$

Gradient Descent: Multivariable Case

Since R is symmetric positive definite

$$R = Q\Lambda Q^T$$

where Q-orthogonal. Therefore,

$$\begin{aligned} y^{[k]} &= (QQ^T - \mu Q\Lambda Q^T)^k y^{[0]} \\ &= Q(I - \mu\Lambda)^k Q^T y^{[0]} \end{aligned}$$

Letting $z = Q^T y$,

$$z^{[k]} = (I - \mu\Lambda)^k z^{[0]} \tag{1}$$

$$\implies z_i^{[k]} = (1 - \mu\lambda_i)^k z_i^{[0]} \tag{2}$$

which converges if $|1 - \mu\lambda_i| < 1$, $i = 1, \ldots, m$.

Gradient Descent: Multivariable Case

Therefore, convergence happens when

$$\begin{aligned} & -1 < 1 - \mu\lambda_i < 1 \\ \iff & -2 < -\mu\lambda_i < 0 \\ \iff & 0 < \mu\lambda_i < 2 \\ \iff & 0 < \mu < \frac{2}{\lambda_i} \end{aligned}$$

Recall that $\lambda_i > 0$ when R is positive definite, so if

$$0 < \alpha < \frac{1}{\lambda_{\max}(R)}$$

then steepest descent converges for quadratic functions.

Gradient Descent: Multivariable Case

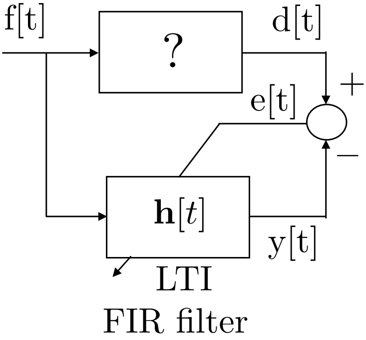
Note that the convergence along each eigenaxis is determined by $\frac{1}{\lambda_i}$.

Therefore if R is ill-conditioned, i.e., $\dfrac{\lambda_{\max}}{\lambda_{\min}}$ is large, then convergence for gradient descent will be much slower along some axes than others.

Section 3

Application: LMS Adaptive Filtering

LMS Adaptive Filtering



Recall the RLS adaptive filter algorithm.
The objective is to minimize the error

$$J(\mathbf{h}) = (d[t] - y[t])^2.$$

- The RLS minimizes the squared error of all past outputs, but LMS only minimizes the squared error of the current output.
- The RLS algorithm was derived using the projection theorem.
- LMS is derived using gradient descent.

LMS Adaptive Filtering

Assume that the output of the adaptive filter is

$$y[t] = \sum_{\ell=0}^{m-1} h[\ell] f[t-\ell] = \mathbf{f}^\top[t]\mathbf{h}$$

where

$$\mathbf{f}[t] = \begin{pmatrix} f[t] \\ f[t-1] \\ \vdots \\ f[t-m+1] \end{pmatrix} \text{ and } \mathbf{h} = \begin{pmatrix} h[0] \\ h[1] \\ \vdots \\ h[m-1] \end{pmatrix}$$

LMS Adaptive Filtering

Then

$$\begin{aligned} J(\mathbf{h}) &= (d[t] - y[t])^2 \\ &= (d[t] - \mathbf{f}^\top[t]\mathbf{h})^2 \\ &= d^2[t] - d[t]\mathbf{f}^\top[t]\mathbf{h} - d[t]\mathbf{h}^\top f[t] + \mathbf{h}\mathbf{f}[t]\mathbf{f}^\top[t]\mathbf{h} \end{aligned}$$

where

$$\frac{\partial J}{\partial \mathbf{h}} = 2\mathbf{f}[t]\mathbf{f}^\top[t]\mathbf{h} - 2d[t]\mathbf{f}[t]$$

LMS Adaptive Filtering

So let

$$\mathbf{h}[t+1] = \mathbf{h}[t] - \alpha \frac{\partial J}{\partial \mathbf{h}}(\mathbf{h}[t])$$

gives

$$\begin{aligned}\mathbf{h}[t+1] &= \mathbf{h}[t] - 2\alpha(\mathbf{f}[t]\mathbf{f}^\top[t]\mathbf{h}[t] - d[t]\mathbf{f}[t]) \\ &= \mathbf{h}[t] + \mu\mathbf{f}[t](d[t] - \mathbf{f}^\top[t]\mathbf{h}[t])\end{aligned}$$

$$\boxed{\mathbf{h}[t+1] = \mathbf{h}[t] + \mu\mathbf{f}[t]e[t]}$$

This is known as the LMS adaptive filter.

Compare to RLS...
For discussion on convergence, consult Moon Chap 14...

Section 4

Gauss-Newton Optimization

Least Squares as a Gradient Descent Problem

Consider the least squares problem

$$\min_{x \in \mathbb{R}^n} \quad \|Ax - b\|_2^2$$

where $A \in \mathbb{R}^{m \times n}$ is tall. We know that the solution is

$$x^* = (A^\top A)^{-1} A^\top b.$$

Can we pose this as a gradient descent problem?

Least Squares as a Gradient Descent Problem

Define the <u>residual</u> as

$$\mathbf{r}(x) = \begin{pmatrix} r_1(x) \\ \vdots \\ r_m(x) \end{pmatrix} = Ax - b$$

and define the sum-of-squares error as

$$\begin{aligned} S(x) &= \frac{1}{2}\mathbf{r}^\top(x)\mathbf{r}(x) \\ &= \frac{1}{2}\sum_{j=1}^{m} r_j^2(x) \\ &= \frac{1}{2}(Ax-b)^\top(Ax-b) \\ &= \frac{1}{2}\left\|Ax-b\right\|_2^2 . \end{aligned}$$

The least squares problem is to find x that minimizes $S(x)$.

Least Squares as a Gradient Descent Problem

The gradient of S is given by

$$\begin{aligned}\frac{\partial S}{\partial x} &= \frac{\partial \mathbf{r}}{\partial x}^{\top}(x)\mathbf{r}(x)\\ &= A^{\top}(Ax - b) = A^{\top}Ax - A^{\top}b.\end{aligned}$$

So the gradient descent algorithm gives

$$x^{[k+1]} = x^{[k]} - \alpha\left(A^{\top}Ax^{[k]} - A^{\top}b\right)$$

In general, we might allow $\alpha > 0$ to be a positive definite matrix $\mathscr{A} > 0$:

$$x^{[k+1]} = x^{[k]} - \mathscr{A}\left(A^{\top}Ax^{[k]} - A^{\top}b\right).$$

Least Squares as a Gradient Descent Problem

Selecting

$$\mathscr{A} = (A^\top A)^{-1}$$

gives

$$\begin{aligned} x^{[k+1]} &= x^{[k]} - (A^\top A)^{-1}\left(A^\top A x^{[k]} - A^\top b\right) \\ &= x^{[k]} - (A^\top A)^{-1}(A^\top A)x^{[k]} + (A^\top A)^{-1}A^\top b \\ &= (A^\top A)^{-1}A^\top b, \end{aligned}$$

which is the optimal solution.
Noting that $A = \frac{\partial \mathbf{r}}{\partial x}$, we have shown that the iteration

$$x^{[k+1]} = x^{[k]} - \left(\frac{\partial \mathbf{r}}{\partial x}^\top(x^{[k]})\frac{\partial \mathbf{r}}{\partial x}(x^{[k]})\right)^{-1}\frac{\partial \mathbf{r}}{\partial x}^\top(x^{[k]})\mathbf{r}(x^{[k]})$$

converges to the optimal in <u>one</u> step when $\mathbf{r}(x) = Ax - b$.

Nonlinear Least Squares

Let $r_j(x)$, $j = 1, \ldots, m$ be a general set of residual function to be minimized. In other words, suppose we wish to solve

$$\min_{x \in \mathbb{R}^n} \quad \frac{1}{2}\mathbf{r}^\top(x)\mathbf{r}(x).$$

Let $\mathbf{J}(x) \triangleq \frac{\partial \mathbf{r}}{\partial x}(x)$. Then the <u>Gauss-Newton</u> (GN) iteration algorithm is given by

$$x^{[k+1]} = x^{[k]} - \left(\mathbf{J}^\top(x^{[k]})\mathbf{J}(x^{[k]})\right)^{-1} \mathbf{J}^\top(x^{[k]})\mathbf{r}(x^{[k]})$$

We know that the GN method converges in one step for the linear least squares problem.

Section 5

Levenberg-Marquardt Optimization

Nonlinear Least Squares

The downside of GN is that the matrix $J^\top(x)J(x)$ may be ill-conditions at some states x.
For the general nonlinear least squares problem, we have

$$\frac{\partial \frac{1}{2}\mathbf{r}^\top(x)\mathbf{r}(x)}{\partial x} = \frac{\partial \mathbf{r}^\top}{\partial x}(x)\mathbf{r}(x) = \mathbf{J}^\top(x)\mathbf{r}(x).$$

Therefore we have

$$\begin{aligned} \text{Gradient Descent} \quad & x^{[k+1]} = x^{[k]} - \alpha \mathbf{J}^\top(x^{[k]})\mathbf{r}(x^{[k]}) \\ \text{Gauss-Newton} \quad & x^{[k+1]} = x^{[k]} - \left(\mathbf{J}^\top(x^{[k]})\mathbf{J}(x^{[k]})\right)^{-1} \mathbf{J}^\top(x^{[k]})\mathbf{r}(x^{[k]}). \end{aligned}$$

Note that there is no inverse for Gradient Descent, but it may converge slowly, even for linear residuals.

Nonlinear Least Squares

The Levenberg-Marquardt (LM) iteration is a combination of gradient descent and Gauss-Newton:

$$x^{[k+1]} = x^{[k]} - \left(\lambda I + \mathbf{J}^\top(x^{[k]})\mathbf{J}(x^{[k]})\right)^{-1} \mathbf{J}^\top(x^{[k]})\mathbf{r}(x^{[k]}),$$

where $\lambda = 1/\alpha$.
Note that $\lambda I + \mathbf{J}^\top\mathbf{J}$ is guaranteed to be full rank and well conditioned for large λ.
Standard practice:

- For the first iteration make λ large (e.g., $\approx 10^4$)
- If squared error decreases, decrease λ for next iteration (e.g., by half).
- If squared error increases, increase λ for next iteration (e.g., by 2x).

Weighted Nonlinear Least Squares

If $W = W^\top > 0$ is a weighting matrix, then

$$\min_{x \in \mathbb{R}^n} \quad \frac{1}{2}\mathbf{r}^\top(x) W \mathbf{r}(x).$$

results in

$$\begin{aligned}
&\text{(GD)} \quad x^{[k+1]} = x^{[k]} - \lambda^{-1}\mathbf{J}^\top W \mathbf{r}\Big|_{x^{[k]}} \\
&\text{(GN)} \quad x^{[k+1]} = x^{[k]} - \left(\mathbf{J}^\top W \mathbf{J}\right)^{-1}\mathbf{J}^\top W \mathbf{r}\Big|_{x^{[k]}} \\
&\text{(LM)} \quad x^{[k+1]} = x^{[k]} - \left(\lambda I + \mathbf{J}^\top W \mathbf{J}\right)^{-1}\mathbf{J}^\top W \mathbf{r}\Big|_{x^{[k]}}.
\end{aligned}$$