☆ Recap with new notations

In gradient descent $x_2 = x_1 - \eta \frac{d}{dx} f(x)$

The above formula is only for a univariate function.

For a bivariate function it will be:

$$x_2 = x_1 - \eta \left(\frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y} \right)$$

New notation: $x_1 \to \theta^1$ & $x_2 \to \theta^2$

$$\theta^{t+1} = \theta^t - \eta \sum \frac{\partial f(x,y)}{\partial \theta}$$

$$\theta_2 = \theta_1 - \eta \cdot \frac{\partial}{\partial x} f(x,y)$$

$$\theta_3 = \theta_2 - \eta \frac{\partial}{\partial y} f(x,y)$$

$$\theta^{t+1} = \theta^t - \eta \frac{\partial}{\partial \theta} f(x_k)$$ where k is a random number between 1 to n

☆ Constrained Optimization Problem

Loss f^n: $$\sum \frac{\vec{w}^T \cdot \vec{x} + w_0}{\|\vec{w}\|} \cdot y_i$$

If $\|\vec{w}\| = 1$ ($\leftarrow$ constraint) then it will make it very easy to calculate differentiation of loss function.

$$\underset{\vec{w},\, w_0}{\operatorname{argmin}} -\sum \frac{\vec{w}^T \cdot \vec{x} + w_0}{\|\vec{w}\|} \cdot y_i \quad \text{s.T.} \quad \|\vec{w}\| = 1$$

Now $\|\vec{\omega}\| = 1$

$$\frac{\partial L}{\partial w_1} = -\frac{\partial}{\partial w_1} \frac{(w_1 x_1 y_1 + \cancel{w_2 x_2 y_2} \cdots + \cancel{w_n x_n} + \cancel{w_0}}{1}$$

$$= -x_1 y_1$$

$$w_j^{t+1} = w_j^t - \eta \left(-\sum x_i y_i\right)$$

$$\frac{\partial L}{\partial w_2} = -x_2 y_2$$

For $j \neq 0$

$$\frac{\partial L}{\partial w} = -\sum x_i y_i$$

For $j=0$ (or for w_0)

$$w_0^{t+1} = w_0^t - \eta\left(-\sum y_i\right)$$

☆ Suppose our function that we want to minimize is $f(x) = x^2 - 3x - 3$ & we put a constraint that our minima must also satisfy $g(x) = x^2 - 2x - 3$

$$\boxed{\therefore \underset{x}{\operatorname{argmin}} \; f(x) \quad \text{s.t. } g(x)} \text{ — (I)}$$

; s.t. = 'such that' or 'subject to'

Goal: To do something so that $f(x)$ is also minimized and constraint is also taken care of.

Solution: Lagrange's multiplier

Instead of computing (I), we will do this:

$$\underset{x,\lambda}{\operatorname{argmin}} \; \underbrace{f(x)}_{\text{Hey! Minimize the } f(x)!} + \lambda \underbrace{g(x)}_{\text{Keep also } g(x) \text{ in mind...}}$$

Lagrange's Multiplier

Why did we convert 'such that ...' form into above form?

<u>Because</u>:

$$\frac{d}{dx}\left(f(x) + \lambda g(x)\right) = \frac{d}{dx} f(x) + \frac{d}{dx} \lambda g(x)$$

$$\frac{d}{dx}\left(f(x) \text{ s.t. } g(x)\right) = \text{Not possible}$$

$\therefore$ Our problem is converted to an 'unconstrained' form.

☆ Generalized form of the previous formula for n constraints $(g_1(x), g_2(x), \ldots, g_n(x))$:

$$\underset{x, \lambda_1, \lambda_2 \ldots, \lambda_n}{\operatorname{argmin}} \quad f(x) + \lambda_1 g_1(x) + \lambda_2 g_2(x) + \ldots + \lambda_n g_n(x)$$

$\therefore$ Doing this for our example, let 'h' be:

$$\boxed{\underset{x,\lambda}{\arg\min}\ f(x) + \lambda g(x) = \underset{x,\lambda}{\arg\min}\ x^2 - 3x - 3 + \lambda x^2 - 2\lambda x - 3\lambda}$$

To minimize this, we need to find its gradient

$$\nabla h = \begin{bmatrix} \frac{\partial}{\partial x} h \\ \\ \frac{\partial}{\partial \lambda} h \end{bmatrix} = \begin{bmatrix} 2x - 3 + 2\lambda x - 2\lambda \\ \\ x^2 - 2x - 3 \end{bmatrix}$$

At minima gradient is 0.

$\therefore \nabla h = 0$

$\therefore$ $2x + 2\lambda x - 2\lambda - 3 = 0$ — (A)

$x^2 - 2x - 3 = 0 \Rightarrow x^2 - 3x + x - 3 = 0 \Rightarrow x(x-3) + 1(x-3) = 0$

$\therefore x = 3$ OR $x = -1$

putting these values in (A)

For $x=3$

$6+6\lambda-2\lambda-3=0$

$4\lambda+3=0$

$\lambda=\frac{-3}{4}$

& For $x=-1$

$-2-2\lambda-2\lambda-3=0$

$-4\lambda-5=0$

$\therefore\ \lambda=\frac{-5}{4}$

The two points are: $\left(3, \frac{-3}{4}\right)$ & $\left(-1, \frac{-5}{4}\right)$

$\downarrow \qquad \downarrow$

Value of $f(x)+\lambda g(x)$: $-3 \qquad 1$

☆ Implementing this technique for our loss function-

Then our $f(x) = -\sum_{i=1}^{n} \frac{\vec{\omega}^T \cdot \vec{x} + \omega_0}{\|\vec{\omega}\|} \cdot y_i$ & our constraint

is $\|\vec{\omega}\| = 1 \Rightarrow g(x) : \|\vec{\omega}\| - 1 = 0$

$$\underset{\vec{\omega}, \lambda}{\operatorname{argmin}} f(x) + \lambda g(x) = -\sum_{i=1}^{n} \vec{\omega}^T \cdot \vec{x} + \omega_0 + \lambda(\|\vec{\omega}\| - 1)$$

Let's find gradient

$$\nabla h = \begin{bmatrix} \frac{\partial}{\partial \vec{\omega}} h \\ \frac{\partial}{\partial \lambda} h \end{bmatrix}$$

First component: $\partial h / \partial \vec{w}$

$$\|\vec{w}\| = \sqrt{w_1^2 + w_2^2 + \ldots + w_n^2}$$

$$= \sqrt{[w_1\ w_2 \ldots w_n] \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}}$$

$$\boxed{\|\vec{w}\| = \sqrt{\vec{w}^T \cdot \vec{w}}}$$

$$\boxed{\frac{\partial}{\partial x} \sqrt{x} = \frac{1}{2\sqrt{x}}}$$

$$\boxed{\frac{\partial}{\partial \vec{x}} \vec{x}^T \cdot \vec{x} = 2\vec{x}}$$

$$\frac{\partial}{\partial w} h = -\sum \frac{\partial}{\partial \vec{w}} \vec{w}^T \cdot \vec{x} + w_0 + \lambda(\|\vec{w}\| - 1)$$

$$= -\sum \vec{x} + 0 + \frac{\partial}{\partial \vec{w}} \lambda(\|\vec{w}\| - 1)$$

$$\frac{\partial}{\partial \vec{w}} \lambda(\|\vec{w}\| - 1) = \lambda \frac{\partial}{\partial \vec{w}} \|\vec{w}\| + 0$$

$$= \lambda \frac{\partial}{\partial \vec{w}} \sqrt{\vec{w}^T \cdot \vec{w}}$$

$$\frac{\partial}{\partial \vec{w}} \lambda(\|\vec{w}\| - 1) = \lambda \frac{1}{2\sqrt{\vec{w}^T \cdot \vec{w}}} \cdot \frac{\partial}{\partial \vec{w}} \vec{w}^T \cdot \vec{w}$$

$$= \lambda \frac{\cancel{2}\vec{w}}{\cancel{2}\sqrt{\vec{w}^T \cdot \vec{w}}}$$

$$= \lambda \frac{\vec{w}}{\|\vec{w}\|}$$

$$\frac{\partial}{\partial \vec{w}} h = -\sum \vec{x} + \lambda \frac{\vec{w}}{\|\vec{w}\|}$$

☆ second component of the gradient is : $\frac{\partial h}{\partial \lambda}$

$$= \frac{\partial}{\partial \lambda}\left(-\sum_{i=1}^{n} \vec{\omega}^T . \vec{x} + \omega_0 + \lambda(||\vec{\omega}|| - 1)\right)$$

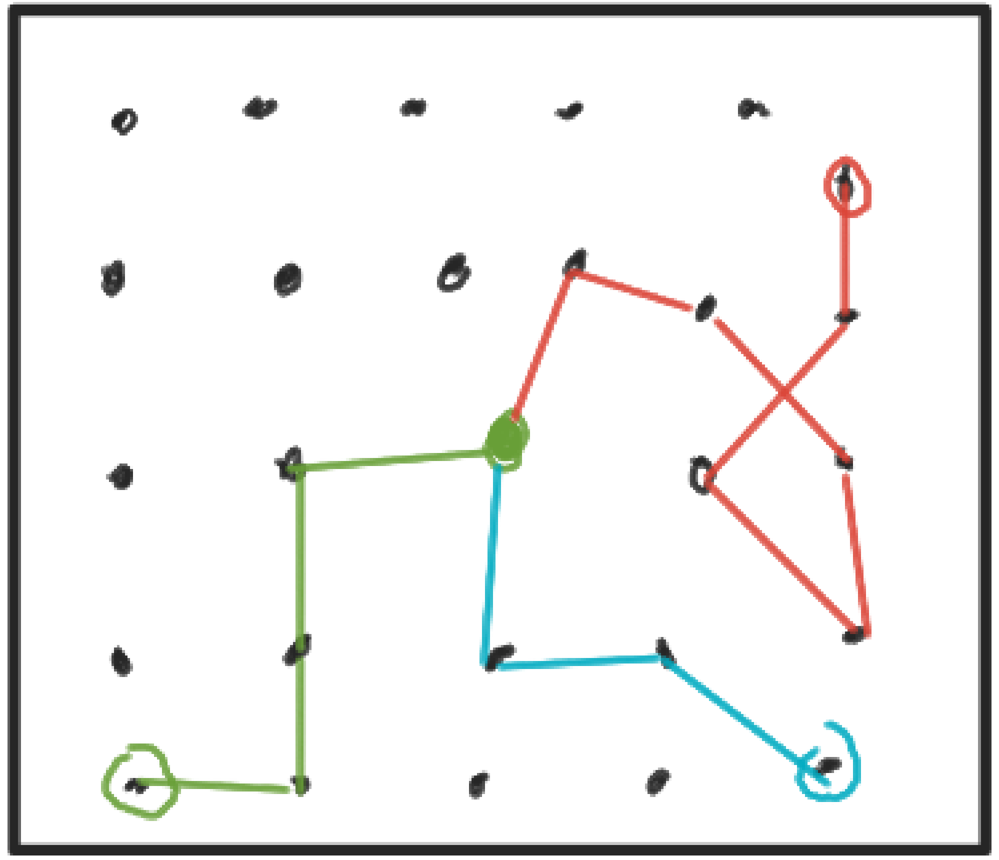
$$= -\sum_{i=1}^{n} 0 + 0 + \frac{\partial}{\partial \lambda} \lambda(||\vec{\omega}|| - 1)$$

$$= -\sum_{i=1}^{n}(||\vec{\omega}|| - 1)$$

☆ Quick Recap :

Gradient Descent : ① Vanilla G.D. | G.D. | Batch G.D.

For finding each next guess it needs N iterations



② Stochastic G.D. - For each next guess, this takes only one iteration. but it will need more guesses to reach to minima.

③ Minibatch G.D. - For each next guess minibatch G.D. takes 'k' iterations where $1 < k < N$.