☆ Now let's implement Lagrange's multiplier technique to minimize our Loss Function of G.D.

$$f(x) = L = -\sum \frac{\bar{\omega}^T \cdot \bar{x} + w_0}{\|\bar{\omega}\|} \cdot y_i \quad s.t. \quad \|\bar{\omega}\| = 1$$

An imp. note: Our constraint must be in the form:

$g(x)$: _____ $= 0$ ∴ Our constraint here will be:

$$g(x): \|\bar{\omega}\| - 1 = 0$$

☆ ∴ Unconstrained form will be:

$$\underset{\bar{\omega}, \lambda}{\arg\min} \; f(x) + \lambda g(x) = \underset{\bar{\omega}, \lambda}{\arg\min} \; -\sum (\bar{\omega}^T \cdot \bar{x} + w_0) \cdot y_i + \lambda(\|\bar{\omega}\| - 1)$$

$\therefore$ gradient $\nabla h = \begin{bmatrix} \frac{\partial}{\partial \bar{w}} h \\ \frac{\partial}{\partial \lambda} h \end{bmatrix}$

$\frac{\partial}{\partial \bar{w}} h \longrightarrow$ First Component

$\frac{\partial}{\partial \lambda} h \longrightarrow$ Second Component

Before calculating these components, let's see some interesting results:

$$\|\bar{w}\| = \sqrt{w_1^2 + w_2^2 + \dots + w_n^2}$$

$$= \sqrt{[w_1, w_2, \dots w_n] \cdot \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}}$$

$$\boxed{\|\bar{w}\| = \sqrt{\bar{w}^t \cdot \bar{w}}} \text{ — I}$$

$$\boxed{\frac{d}{dx}\sqrt{x} = \frac{1}{2\sqrt{x}}} \text{ — II}$$

$$\boxed{\frac{\partial}{\partial \bar{x}} \bar{x}^t \cdot \bar{x} = 2\bar{x}} \text{ — III}$$

★ First Component : $\frac{\partial}{\partial \bar{\omega}} h = \frac{\partial}{\partial \bar{\omega}}\left(-\sum(\bar{\omega}^T.\bar{x} + \omega_0)y_i + \lambda(\|\bar{\omega}\|-1)\right)$

$$= \underset{\bar{\omega},\lambda}{\operatorname{argmin}} \; -\sum y_i(\bar{x}+0) + \frac{\partial}{\partial \bar{\omega}}\lambda\|\bar{\omega}\| - 0$$

$$= \underset{\bar{\omega},\lambda}{\operatorname{argmin}} \; -\sum y_i \cdot \bar{x} + \lambda\frac{\partial}{\partial \bar{\omega}}\left(\sqrt{\bar{\omega}^t.\bar{\omega}}\right) \quad (\text{From I})$$

$$= \underset{\bar{\omega},\lambda}{\operatorname{argmin}} \; -\sum y_i\bar{x} + \frac{\lambda}{2\sqrt{\bar{\omega}^t.\bar{\omega}}} \cdot \frac{\partial}{\partial \bar{\omega}}(\bar{\omega}^t.\bar{\omega}) \quad (\text{From II})$$

$$= \underset{\bar{\omega},\lambda}{\operatorname{argmin}} \; -\sum y_i\bar{x} + \cancel{2}\lambda\bar{\omega} / \cancel{2}\sqrt{\bar{\omega}^t.\bar{\omega}}$$

$$\boxed{\frac{\partial}{\partial \bar{\omega}} h = \underset{\bar{\omega},\lambda}{\operatorname{argmin}} \; -\sum y_i \cdot \bar{x} + \lambda \frac{\bar{\omega}}{\|\bar{\omega}\|}}$$

☆ Second Component $= \frac{\partial}{\partial \lambda} h = \frac{\partial}{\partial \lambda} \lambda \cdot (\|\bar{\omega}\| - 1)$

$$= \frac{\partial}{\partial \lambda} \lambda \|\bar{\omega}\| - \frac{\partial}{\partial \lambda} \lambda \quad = \quad \|\bar{\omega}\| - 1$$

☆ 3-variants of G.D. -

① Vanilla G.D. / Batch G.D: $w_i^{t+1} = w_i^t - \eta(\nabla f)$

This will take us to the next guess from the initial guess. Suppose we need take 1000 iterations of taking next guess to reach the minima then let's call $n = 1000$.
In each iteration to compute ∇f, we need to differentiate f 'd' no. of times where d = dimensions.
In each differentiation, we need to go throug each x_i & y_i

So let $m = 10,000$ be our no. of datapoints. Hence to take each new guess Vanilla G.D. will do $m \cdot d$ computations ($10,000 \times 50 = 50,000$; assuming $d = 50$)

As we see it is computationally very expensive.

(2) Stochastic G.D. - This variant of GD doesn't consider all 'm' datapoints to find the next guess but only 1 random datapoint.

As a result, this will need more iterations to converge.

(3) Minibatch G.D. - This considers k datapoints (k can be any number) to compute the next guess.

As per CLT, the mean of k datapoints is close to actual mean hence, this variant of GD will have less randomness than Stochastic & hence will need less iterations than it.