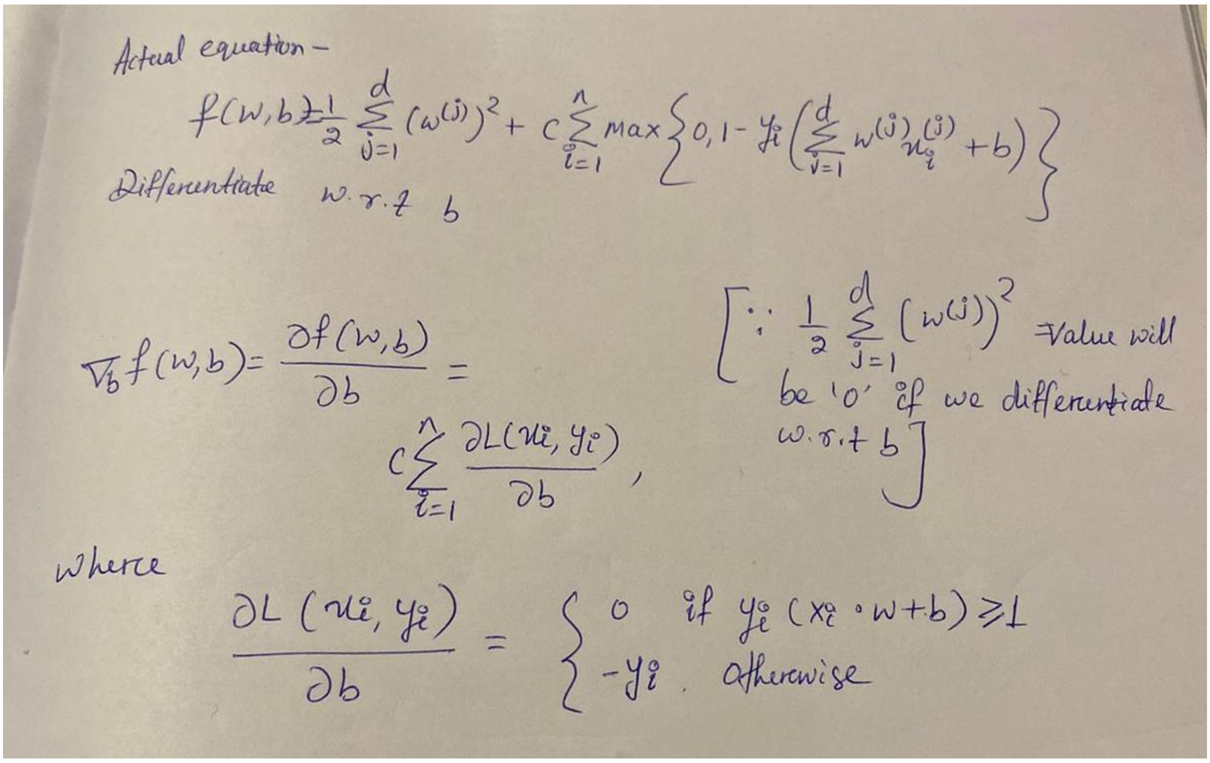


Actual equation –

$$f(w,b) = \frac{1}{2}\sum_{j=1}^{d}(w^{(j)})^2 + C\sum_{i=1}^{n}\max\left\{0, 1 - y_i\left(\sum_{j=1}^{d} w^{(j)} x_i^{(j)} + b\right)\right\}$$

Differentiate w.r.t b

$$\nabla_b f(w,b) = \frac{\partial f(w,b)}{\partial b} = C\sum_{i=1}^{n}\frac{\partial L(x_i, y_i)}{\partial b},$$

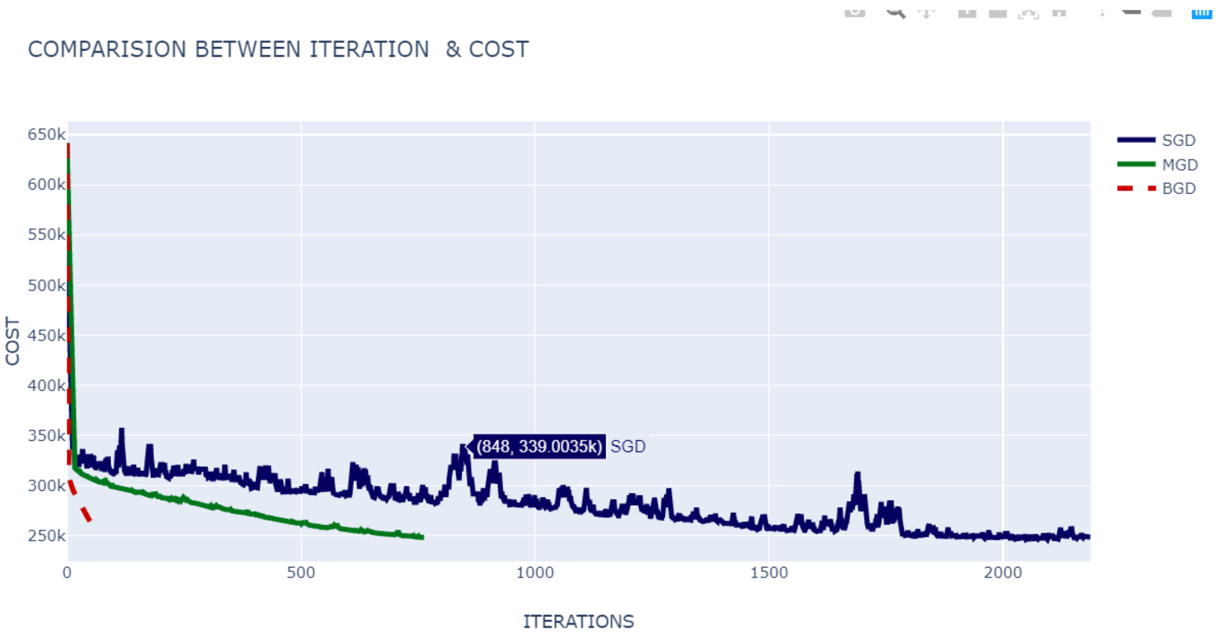
$\left[\because \frac{1}{2}\sum_{j=1}^{d}(w(j))^2\right.$ = Value will be '0' if we differentiate w.r.t b$\left.\right]$

where

$$\frac{\partial L(x_i, y_i)}{\partial b} = \begin{cases} 0 & \text{if } y_i(x_i \cdot w + b) \geq 1 \\ -y_i & \text{otherwise} \end{cases}$$

Gradients:

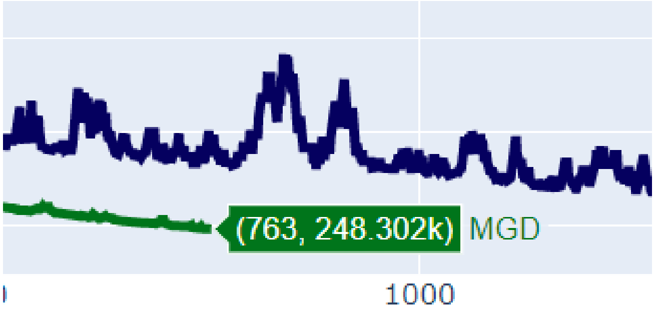
$$\nabla_b f(\mathbf{w}, b) = C\sum_{i=1}^{n}\frac{\partial M}{\partial w^{(j)}} \qquad \frac{\partial M}{\partial w^{(j)}} = \begin{cases} 0, & y_i(\mathbf{x_i}\cdot\mathbf{w} + b) \geq 1 \\ -y_i, & y_i(\mathbf{x_i}\cdot\mathbf{w} + b) < 1 \end{cases}$$



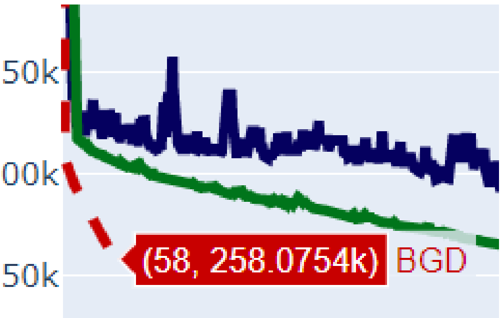
This is the sample run time

Interpretation:

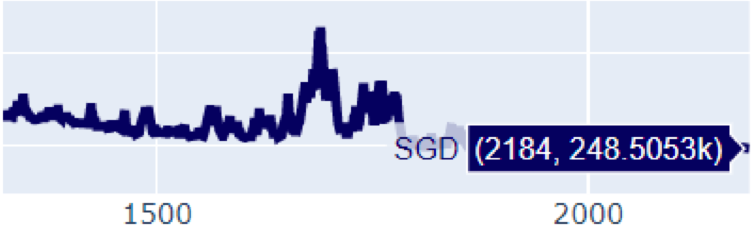
Convergence time for mini-batch gradient descent(MGD) is 763



Convergence time for batch gradient descent(BGD) is 58, 258.0754k cost



Convergence time for stochastic gradient descent(SGD) is 2184



From the plot we saw that BGD takes the least number of iterations to converge. Batch Gradient Descent is great for convex or relatively smooth

But while executing, every iteration for BGD took more time than SGD and MGD

SGD seems to be more NOISY. This is because of the effects which we are taking from 6414 samples. There is a chance that sample is having Noise but again SGD can be very much useful for Deep Learning as we will consider in terms of batch size and Epochs which help more to get a smoother curve.

Moreover the SGD is kind pf acting as a **Impeller** which helps to push the gradient descent out of local minimum values of the cost function. This is the reason that SGD out performed after 2184 iterations.

SGD, the average cost over the epochs in MGD fluctuates less because we are averaging less samples at a time.

However, stochastic and MGD demonstrate saturation, and cost function can less likely to be decreased with the increase of the number of iterations.

On the opposite side, BGD able to decrease the cost function after the stage obtained of quick descent with slower rate.

Conclusion:

SGD and MGD methods permit to quicker descent to the neighbourhood of minimum of cost function and get saturated after certain iteration.

BGD may take longer time, the good thing is it decreases the cost function more than SGD and MGD