1. Thortical Part

1.1. convex optimzation

1. by definition of convex function $g: C \to \mathbb{R}$ is a convex if for all $u, v \in C$ and $\alpha \in [0,1]$

$\Rightarrow g(\alpha u + (1-\alpha)v) \leq \alpha g(u) + (1-\alpha) g(v)$

if $g(u) = \sum_{i=1}^{m} \gamma_i f_i$

$$g(\alpha u + (1-\alpha)v) = \sum_{i=1}^{m} \gamma_i f_i(\alpha u + (1-\alpha)v) \leq \sum_{i=1}^{m} \gamma_i \left(\alpha f_i(u) + (1-\alpha) f_i(v)\right)$$

We know that f_i convex for every i

$$= \sum_{i=1}^{m} \gamma_i \alpha f_i(u) + \gamma_i (1-\alpha) f_i(v) = \sum_{i=1}^{m} \gamma_i \alpha f_i(u) + \sum_{i=1}^{m} \gamma_i (1-\alpha) f_i(v)$$

$$= \alpha g(u) + (1-\alpha) g(v) \quad \blacksquare$$

2. $f(x) = x^2$, $g(x) = x^2 - 1$, $h(x) = (x^2-1)^2 = x^4 - 2x^2 + 1$

$h'(x) = 4x^3 - 4x$

$h''(x) = 12x^2 - 4$

The second derivatives is not always positive hance isnt convex

1.2 Sub-gradients for Soft svm objective

3. We got 2 function $f_1 = 0$ $f_2(x) = 1 - y(x^T w + b)$

f_1 is convex - trival

f_2 is function - its linear function $\Rightarrow$ convex

in the lection we told thats the sup of 2 convex functins is convex

$\Rightarrow f(w, b)$ is convex

4. if the class correct then $f(w,b) = 0$

in 3 we prove that f is convex and in the lection we saw that $0 \in \partial f(x) \Rightarrow 0$ is global min and in convex function local min is global.

so $0 \in \partial f(x)$

5. In the lecture we saw that

$$\partial(f_1+f_2) = \partial(f_1)+\partial(f_2) = \{u+v \mid u\in\partial f_1, v\in\partial f_2\}$$

thus if $g_k \in \partial f_k \Rightarrow \sum_{k=1}^{n} g_k \in \partial \sum_{k=1}^{n} f_k$

6. for every $w\in\mathbb{R}^d$ s.t $\gamma_{x_i,y_i}(w,b)$ if we

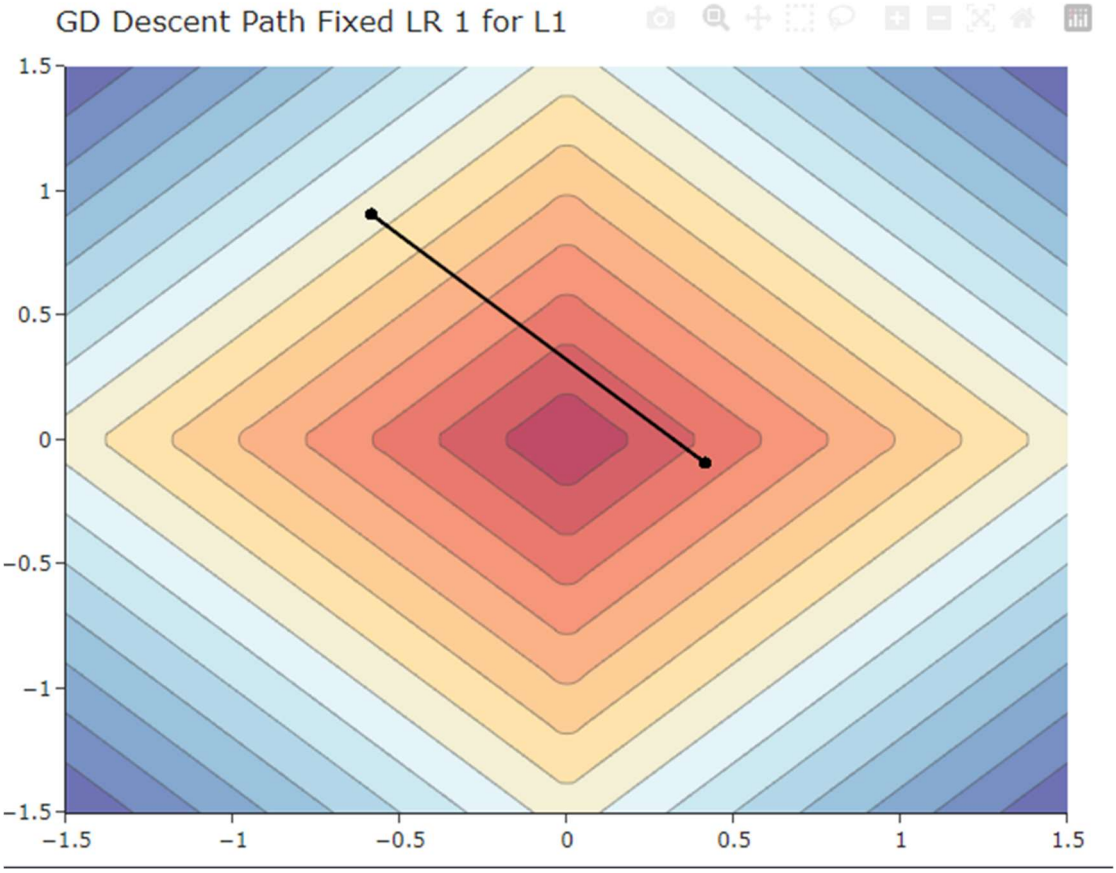
sum over $\sum_{i=1}^{n} \gamma_{x_i,y_i}(w,b)$ from 5 we know

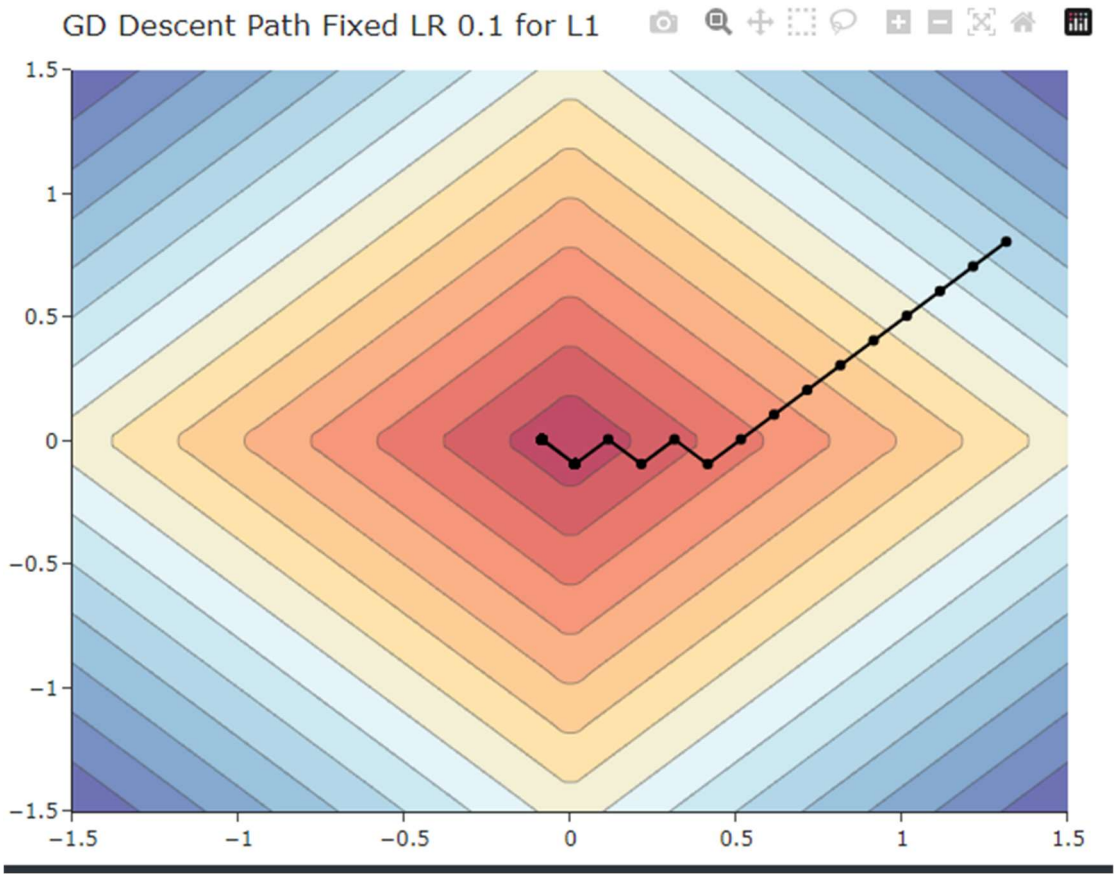
that it is equal to $g\in\partial f(w,b) = g_i \in \partial \gamma_{x_i,y_i}(w,b)$

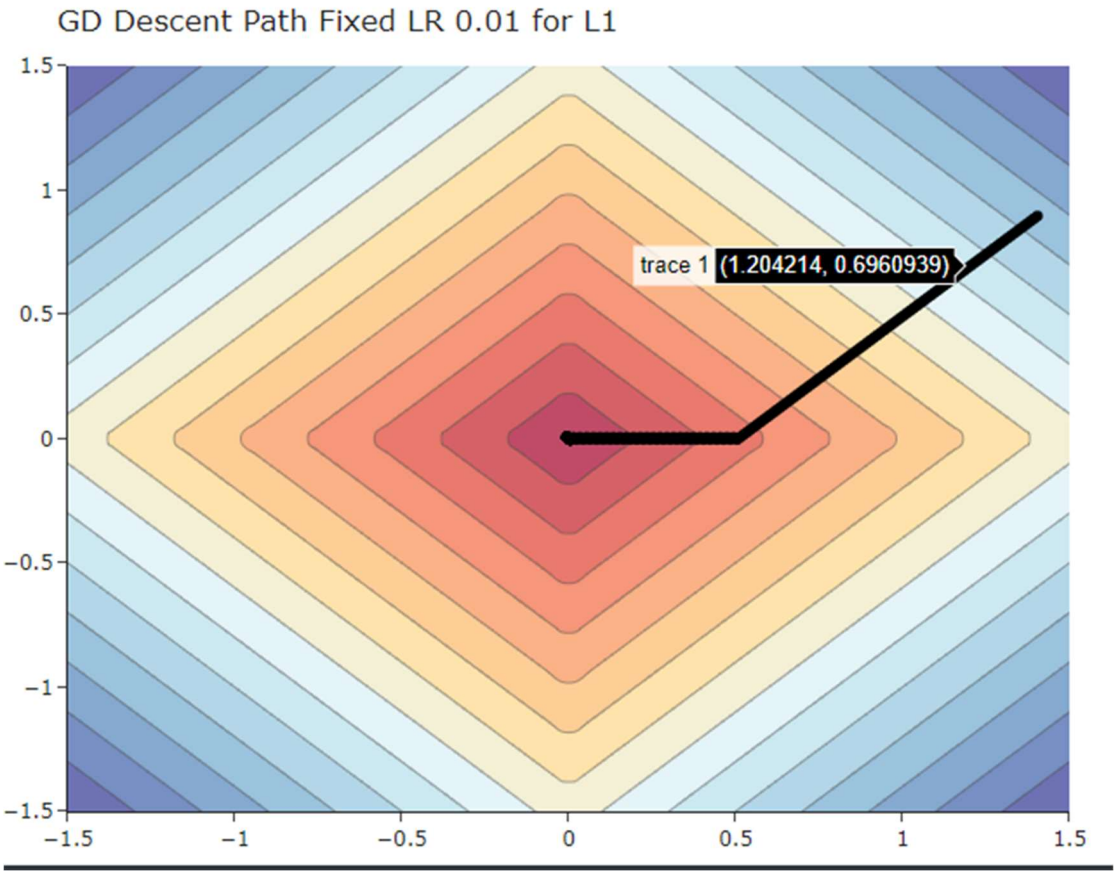
for w that holds for $f(w,b)=0 \Rightarrow 0\in\partial f(x)$ (3)

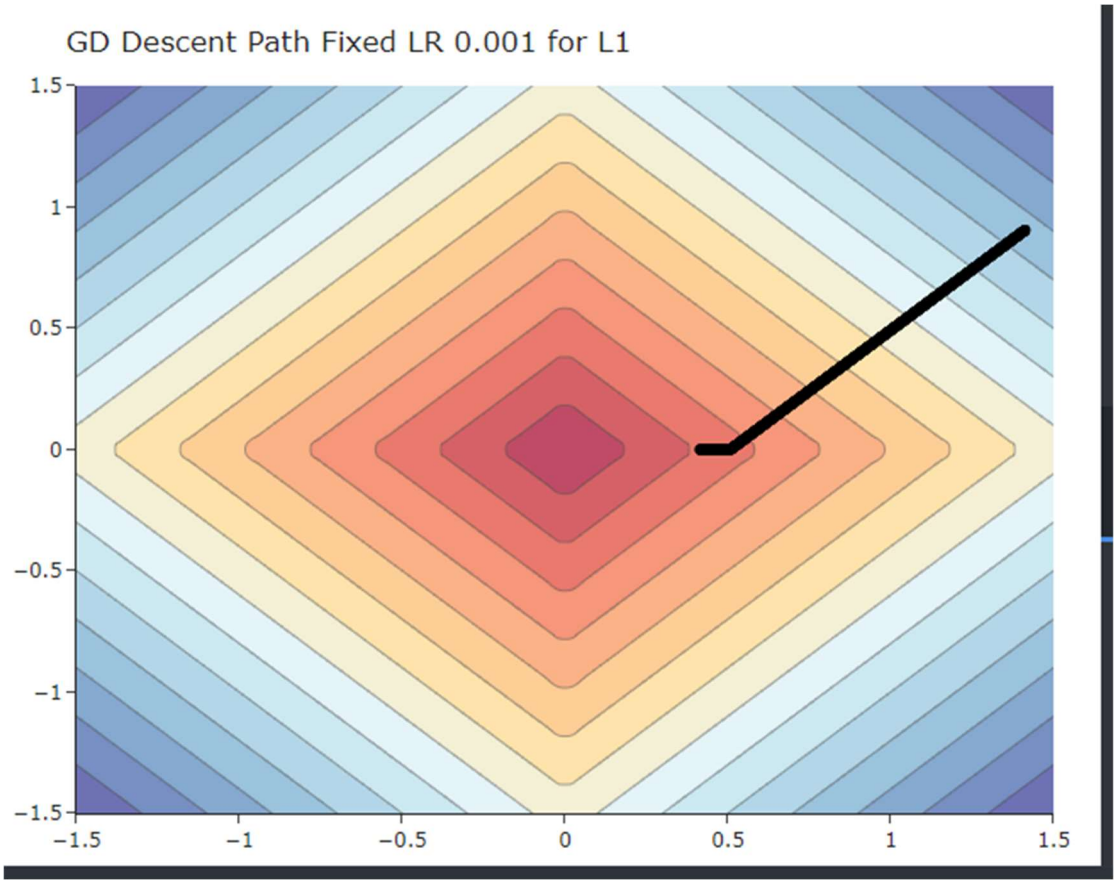
For w s.t $f(w,b)\neq\{0,1\}$ f is differentiable
as sum of diff and it derivative is $-yx \in \partial f(x)$

for w s.t $f_{x_i,y_i}(w,b)=0$ and $(1-x_i^T w+b)<0$ the margin
is not differentiable this is like
the absolut value function so the sub
gradient is between the value of the
derivitive $\Rightarrow \alpha(-y_i x_i)\in\partial f(x)$

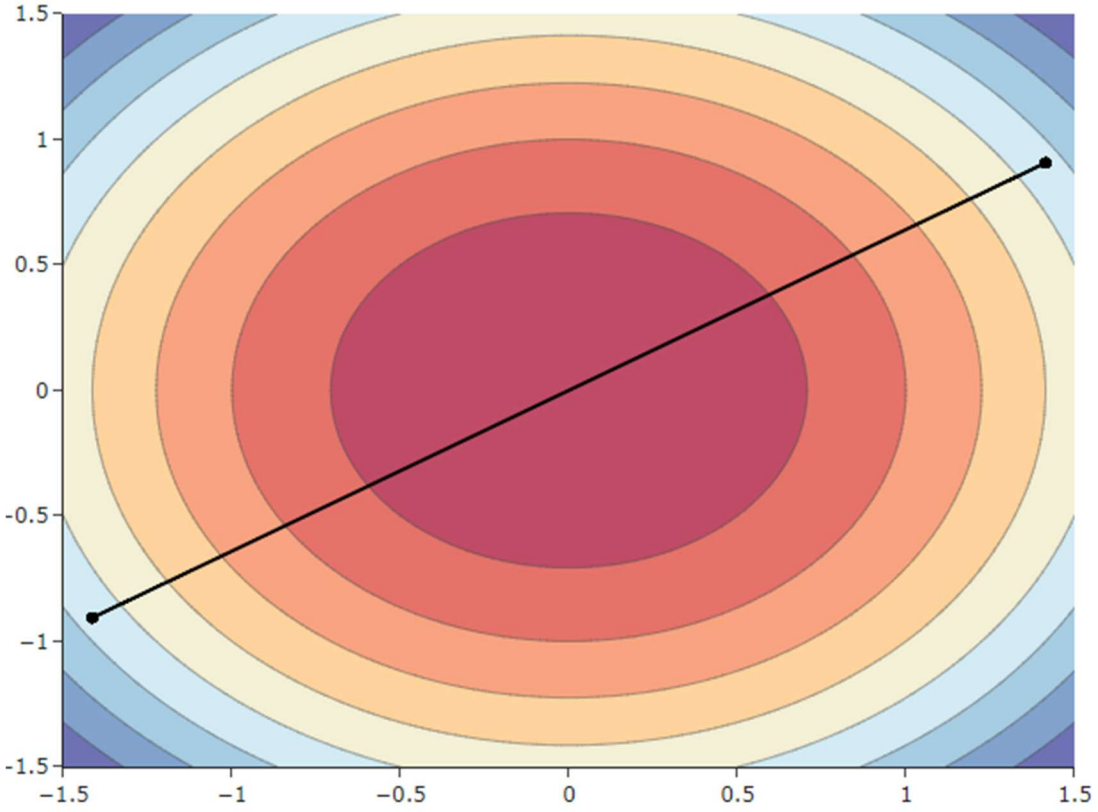

GD Descent Path Fixed LR 1 for L1
1.5
1
0.5
0
−0.5
−1
−1.5
−1.5
−1
−0.5
0
0.5
1
1.5


GD Descent Path Fixed LR 0.1 for L1
1.5
1
0.5
0
−0.5
−1
−1.5
−1.5
−1
−0.5
0
0.5
1
1.5

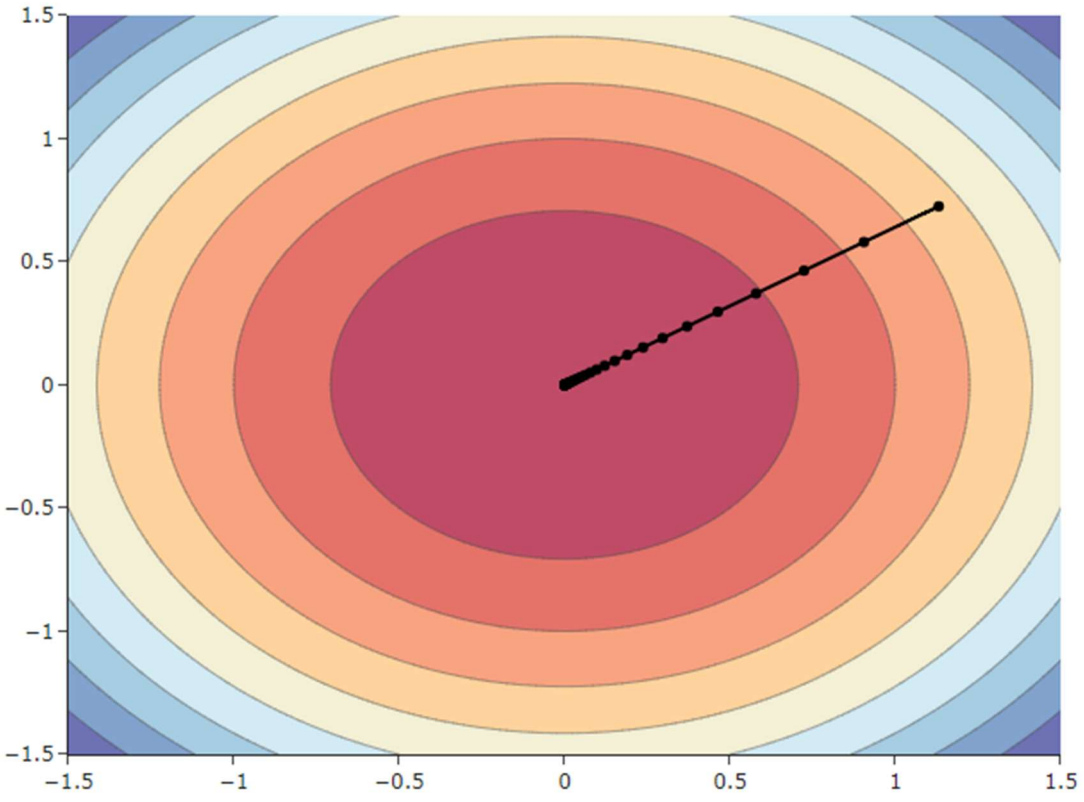

GD Descent Path Fixed LR 0.01 for L1
trace 1 (1.204214, 0.6960939)
1.5
1
0.5
0
−0.5
−1
−1.5
−1.5
−1
−0.5
0
0.5
1
1.5

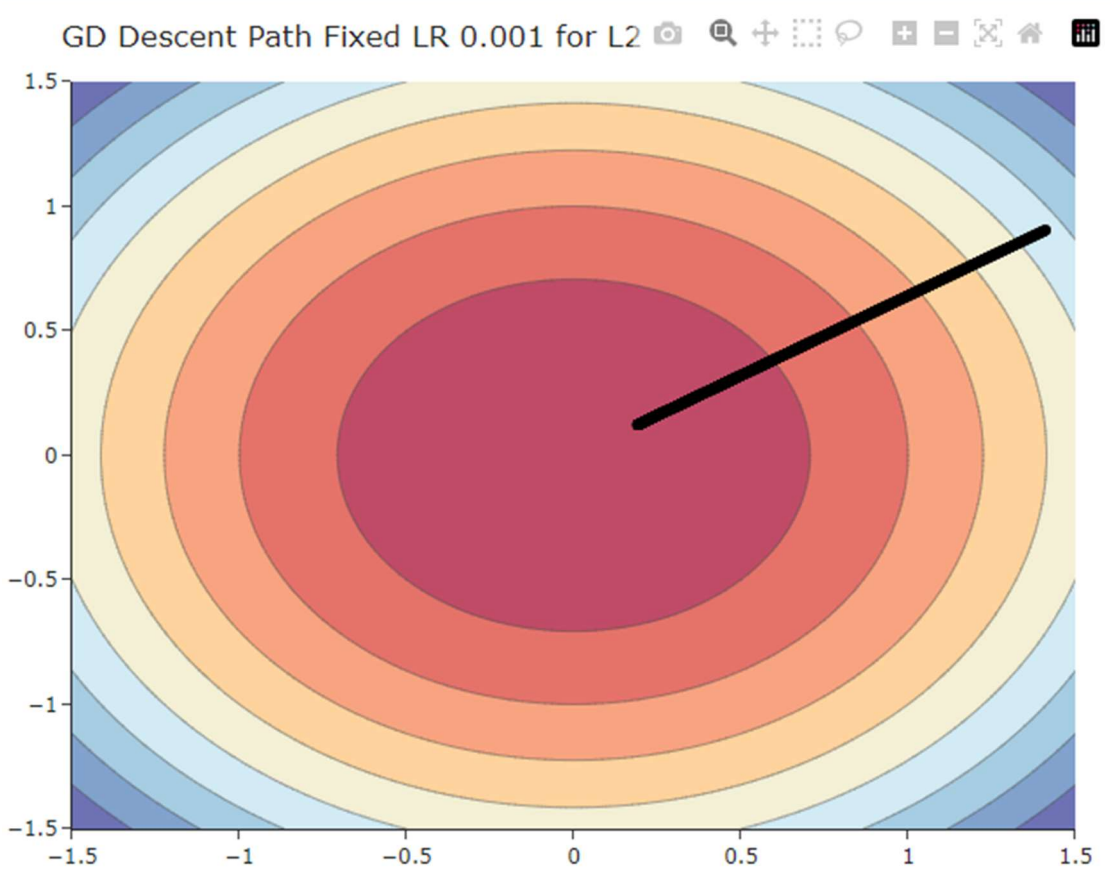

GD Descent Path Fixed LR 0.001 for L1
1.5
1
0.5
0
−0.5
−1
−1.5
−1.5
−1
−0.5
0
0.5
1
1.5

GD Descent Path Fixed LR 1 for L2

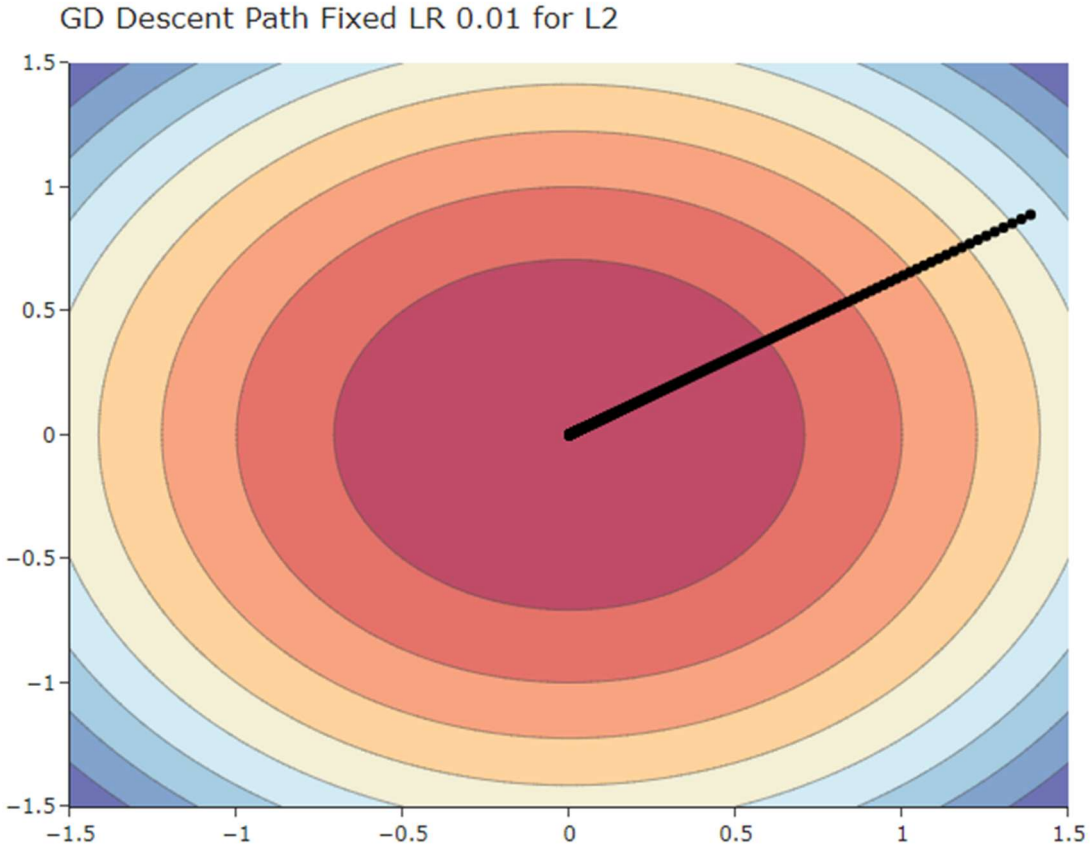


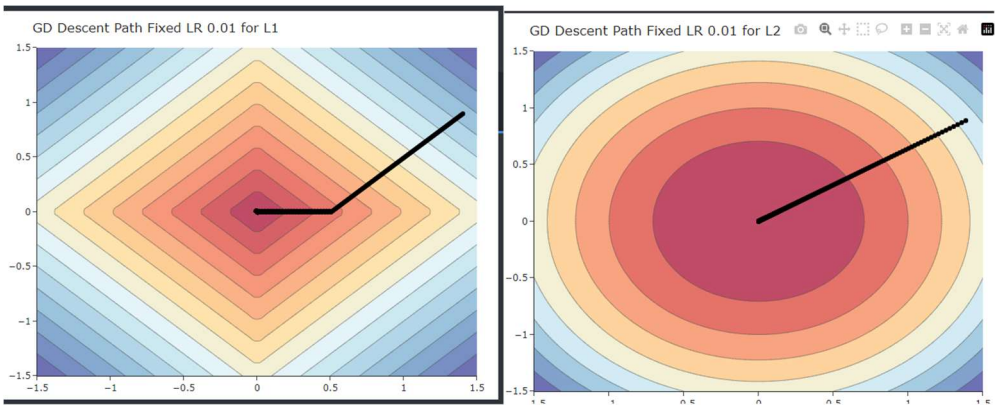
GD Descent Path Fixed LR 0.1 for L2



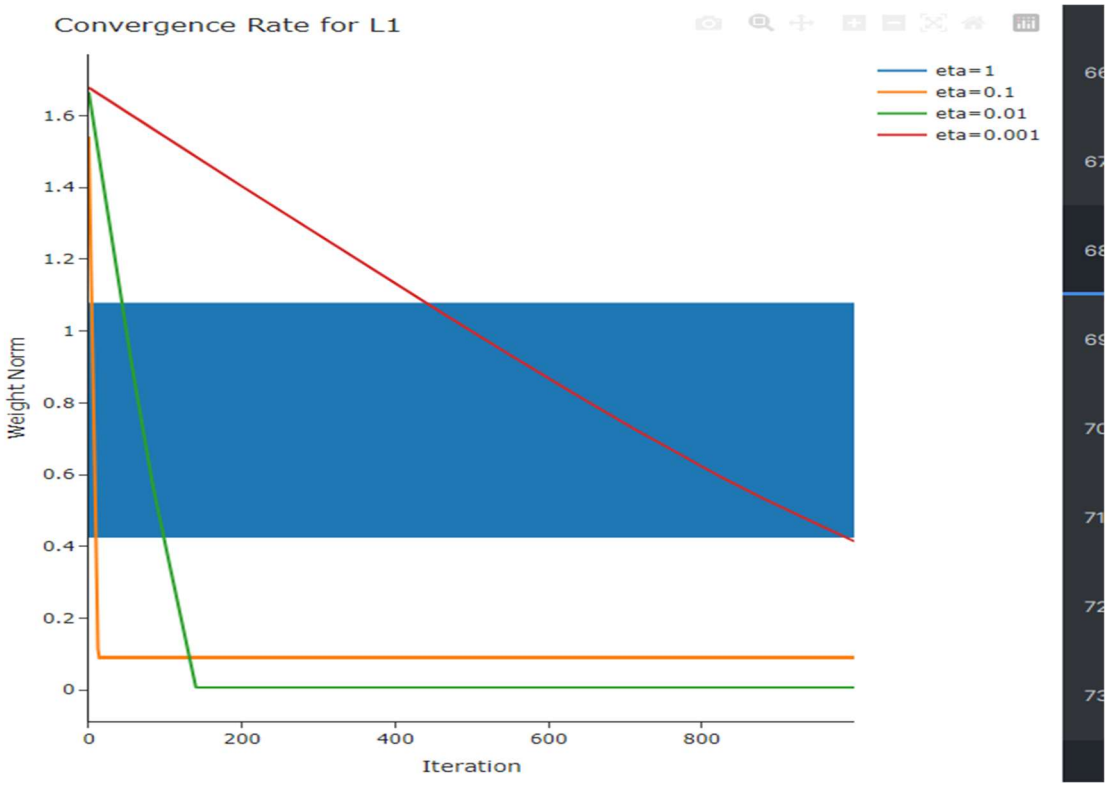

GD Descent Path Fixed LR 0.001 for L2
1.5
1
0.5
0
−0.5
−1
−1.5
−1.5
−1
−0.5
0
0.5
1
1.5

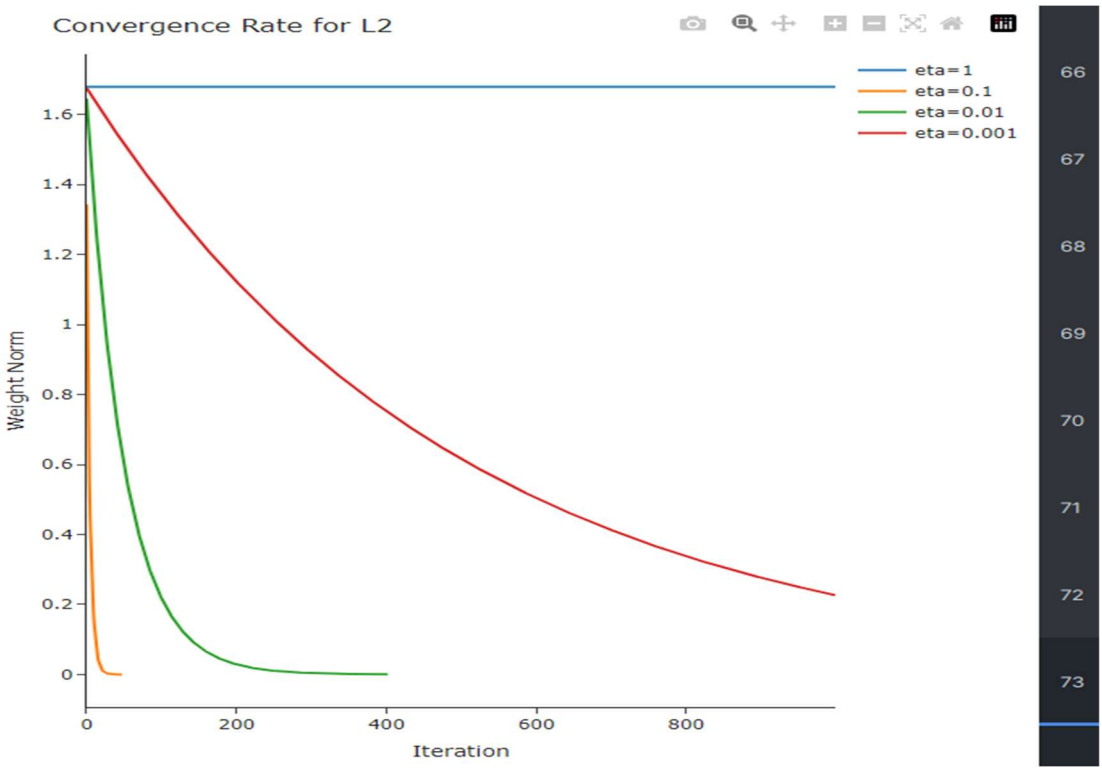
Plots


GD Descent Path Fixed LR 0.01 for L2
1.5
1
0.5
0
−0.5
−1
−1.5
−1.5
−1
−0.5
0
0.5
1
1.5



1. In L1 because when it reaches a corner its change his direction that because it's a square and if it start somewhere else the line should be different but for norm 2 for every starting point it would be the same
2. In the eta 1 in l1 the algorithm took only on step that's because of the condition saying that if the tol gt delta and also when eta is very small the algorithm never achieved minimum
3.





For L1 eta 1 is jumping between this 2 values thus it look like a fool rectangle
eta = 0.1 converge very fast but never get the minimum cause it too large
eta = 0.01is slower but more accurate
eta = 0.001 never get to the min because the step are too small and its done 1000 iteration before finished
For L2 eta = 1 is always at the same distance
eta=0.1 converge very fast to 0 because as we closer to the minimum the gradient is smaller which cause the steps to be small and enable us to find the min
eta = 0.01 like 0.1 but shorter steps
eta = 0.001 never get to the min because the step are too small and its done 1000 iteration before finished

4.

[1.49188038] 1

[0.10811962] 0.1

[0.01188038] 0.01

[0.41430751] 0.001

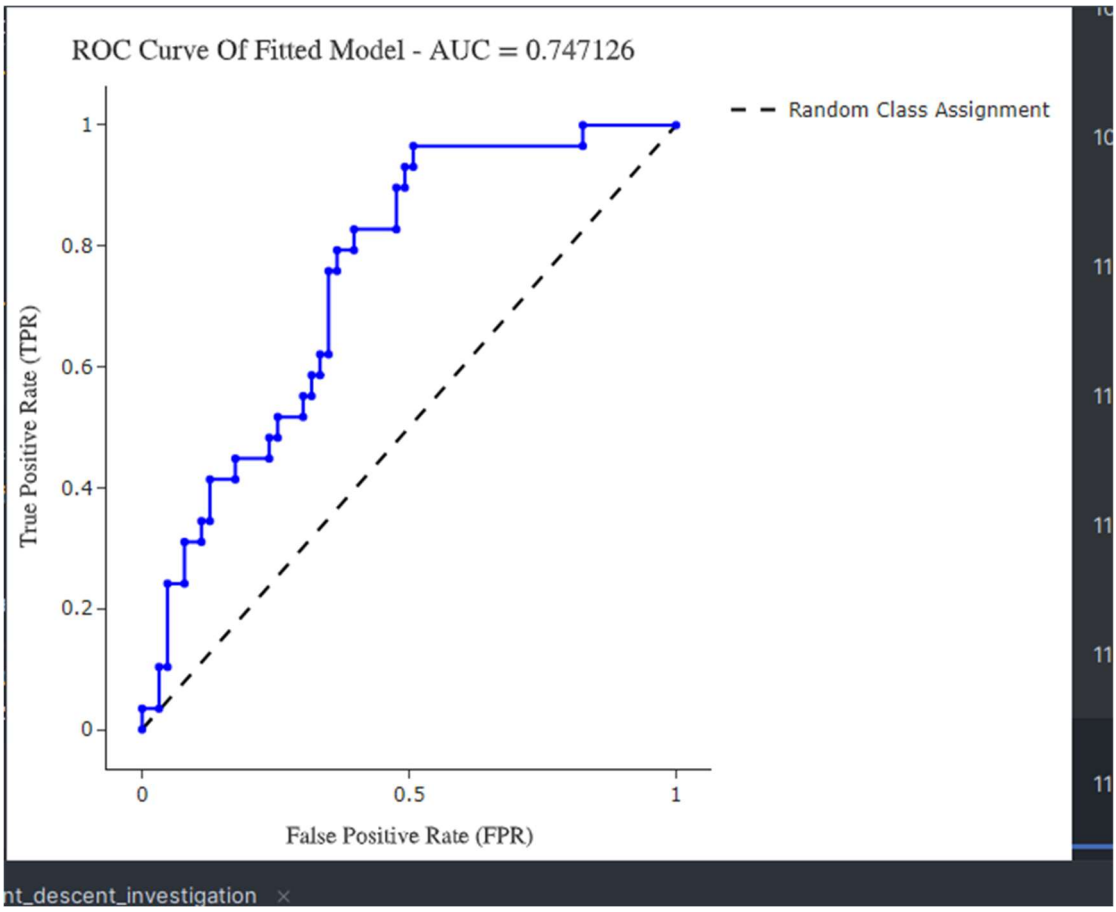
[2.82100623] 1

[e-091.40295195] 0.1

[e-072.39122837] 0.01

[0.051462] 0.001
the difference is because all of the above that was explain in 3

.5



.6

0.0004926617378555309 = alpha

0.358695652173913 = err

7. lambda=0.02
test_err = 0.283