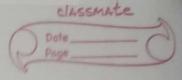


((0.25, 0.5), 1 $y(\omega x+b)=(0.25,0.5)(0.5,-0.5)-$ = 0.125 -0.25-1<0 m 7 Uplate W= (0.75,0 New algorithm 7 Point 3: ((-0.125, 0.5), -1) y(wx+b)=-1(c0.75,0) (-0.125,0.5) 0.15 x 0.125 70 w = (0.75,0) + (0.125, -0.5) if nouls =(0.875, -0.5)(0.0625, 0.5), y (wx+b) = (0.875, -0.5) (0.0625, 0.5) Formetake Ha mitatorit no whole W = (0.815, -0.5) + (0.0625 + 0.5)= (0.94,0) For point 5, b=0 & y=-1 son wx <0 ywx70 and infinite of mistake since built shift between Oand Tresulting in alternating trace corrects and mie takes.



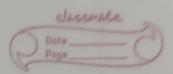
small margin halfshow and a seperate example where it of the dataset be + ((-0.5,0), (C-0.625),0), ((-0.03125,0)  $+1 \rightarrow ((0.25,0), (0.0625,0), (0.016,0)$ Since perceptoron treats Similar to the case in 0.1. Since this is the original algorithm, it will not make infinite mistakes. Instead since points of appaining classes move closes to the x=0 line, the only possible margin halfshore will be arbitrarily small since decision boundary will be a close to x=0 allowing only little charges resulting in a simply of  $0 < \epsilon < \frac{1}{2}$ tor a maximized margin halfshore, Let us use dataset, +1 -> (1,0), (2,0),(3,0)  $-1 \rightarrow (-1,0), (-2,0), (-3,0)$ Here alternating points of the 2 classes can maximizes the margin hollshore.

Let us consider the herselton algorithm in this case

Point 1: ((1,0); 1)

Point 1: ((1,0),1)  $y(\omega x + b) = 10+0 = 0 \le 0$   $y(\omega x + b) = 10+0$  $y(\omega x + b) = 10$ 

 $\begin{array}{c} \text{Point 2: } ((-1,0),1) \\ \text{y (w x + b)} = -1((1,0)(-1,0)+1) \\ -1(1) \leq 0 \end{array}$ 



Point 3: (2,0),1) $\begin{array}{c} (\omega)(+b) = (1((2,0)(2,0)+0) = 4) 6 \\ (-2,0)(-1) \\ -1((2,0)(-2,0)) = 4) \\ \end{array}$ Point 5: ((3,0), 1) 1((2,0)(3,0)) = 670 lout 6: ((-3,0),-1)  $\omega = (2,0)(-3,0) = 670$   $\omega = (2,0) b = 0$  maximizes the forestron margin holfstore for given data while b=0 passes through origin, first data points margin is 1 3. Show how ferrefteron can be viewed as an instantiation Corception is a specific instance of SGD where we and bupdates seem like changes in gradient with iterations Let us consider an arbitrary data point: (x, x2)
If we were to consider the perceptron algorithm, it
would look like: if y (w x + b) < 5 Hen This is what stochastic gradient needs to look like he best loss function in this case for briary classification would be a trage this loss function which will be defined later.

