WARM-UP :

1) Convex function $f: \mathbb{R}^n \to \mathbb{R}$ s.t for $\lambda \in [0,1]$, $x, y \in \mathbb{R}^n$

$$\lambda f(x) + (1-\lambda) f(y) \geq f(\lambda x + (1-\lambda) y)$$

(a) $f(x) = \omega f_1(x)$, S.T f is convex iff f_1 is convex

$\Rightarrow$ Let $x_1, x_2 \in \mathbb{R}^n$

$\Rightarrow$ $f(x_1) = \omega f_1(x_1)$, $f(x_2) = \omega f_2(x_2)$

$$\lambda f(x_1) + (1-\lambda) f(x_2) \overset{?}{\geq} f(\lambda x_1 + (1-\lambda) x_2)$$

$$\Rightarrow \omega \lambda f_1(x_1) + (1-\lambda) \omega f_1(x_2) \overset{?}{\geq} \omega f_1(\lambda x_1 + (1-\lambda) x_2)$$

$$\Rightarrow \lambda f_1(x_1) + (1-\lambda) f_1(x_2) \overset{?}{\geq} f_1(\lambda x_1 + (1-\lambda) x_2) \quad -(1)$$

As f_1 is convex (1) is true $\Rightarrow$ f is also convex

(b) $f(x) = f_1(x) + f_2(x)$, S.T f is convex if f_1 & f_2 are convex

$$\Rightarrow \lambda f(x_1) + (1-\lambda) f(x_2) \overset{?}{\geq} f(\lambda x_1 + (1-\lambda) x_2)$$

$$\Rightarrow \lambda (f_1(x_1) + f_2(x_1)) + (1-\lambda)(f_1(x_2) + f_2(x_2))$$

$$= [\lambda f_1(x_1) + (1-\lambda) f_1(x_2)] + [\lambda f_2(x_1) + (1-\lambda) f_2(x_2)]$$

As f_1, f_2 are convex

$$\lambda f_1(x_1) + (1-\lambda) f_1(x_2) \geq f_1(\lambda x_1 + (1-\lambda) x_2)$$

$$(+) \quad \lambda f_2(x_1) + (1-\lambda) f_2(x_2) \geq f_2(\lambda x_1 + (1-\lambda) x_2)$$

$$[\lambda f_1(x_1) + (1-\lambda) f_1(x_2)] + [\lambda f_2(x_1) + (1-\lambda) f_2(x_1)] \geq f_1(\lambda x_1 + (1-\lambda) x_2) + f_2(\lambda x_1 + (1-\lambda) x_2)$$

$$\lambda f(x_1) + (1-\lambda) f(x_2) \geq f(\lambda x_1 + (1-\lambda) x_2)$$

$\Rightarrow$ f is a convex function

(c) $f(x) = \max\{f_1(x), f_2(x)\}$ **iff** f is convex **iff** f_1, f_2 are convex functions

T.B.P $\lambda f(x_1) + (1-\lambda) f(x_2) \geq f(\lambda x_1 + (1-\lambda) x_2)$

Given: $\lambda f_1(x_1) + (1-\lambda) f_1(x_2) \geq f_1(\lambda x_1 + (1-\lambda) x_2)$ & $\lambda f_2(x_1) + (1-\lambda) f_2(x_2) \geq f_2(\lambda x_1 + (1-\lambda) x_2)$

Without loss of generality.

Let $x_1 \in$ region where $f_1(x) > f_2(x)$ & $x_2 \in$ region where $f_2(x) > f_1(x)$

$\Rightarrow \lambda f(x_1) + (1-\lambda) f(x_2)$

$= \lambda f_1(x_1) + (1-\lambda) f_2(x_2) \geq \lambda f_1(x_1) + (1-\lambda) f_1(x_2) \geq f_1(\lambda x_1 + (1-\lambda) x_2)$

$\geq \lambda f_2(x_1) + (1-\lambda) f_2(x_2) \geq f_2(\lambda x_1 + (1-\lambda) x_2)$

$\Rightarrow f(\lambda x_1 + (1-\lambda) x_2) = \max\{f_1(\lambda x_1 + (1-\lambda) x_2), f_2(\lambda x_1 + (1-\lambda) x_2)\}$

As $\lambda f(x_1) + (1-\lambda) f(x_2) \geq f_1(\lambda x_1 + (1-\lambda) x_2)$ &

$\lambda f(x_1) + (1-\lambda) f(x_2) \geq f_2(\lambda x_1 + (1-\lambda) x_2)$

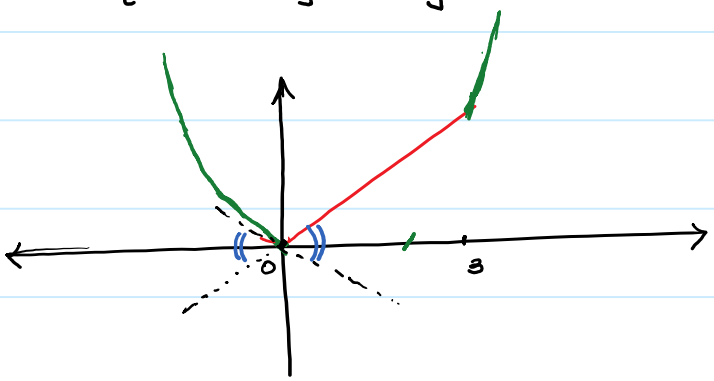
$\Rightarrow \lambda f(x_1) + (1-\lambda) f(x_2) \geq \max\{f_1(\lambda x_1 + (1-\lambda) x_2), f_2(\lambda x_1 + (1-\lambda) x_2)\}$

$\Rightarrow \lambda f(x_1) + (1-\lambda) f(x_2) \geq f(\lambda x_1 + (1-\lambda) x_2)$

So, $f(x) = \max\{f_1(x), f_2(x)\}$ is a convex function

2)

a) $f(x) = \max\{x^2-2x, |x|\}$ Sub gradient @ $x=0, x=-2$



$x>0$, $x^2-2x = x$

$x=0, x=3$

$x \leq 0$, $x^2-2x = -x$

$x=0, x=1$

@ $x=0$ → Non differentiable

@ $x=-2$ → Differentiable

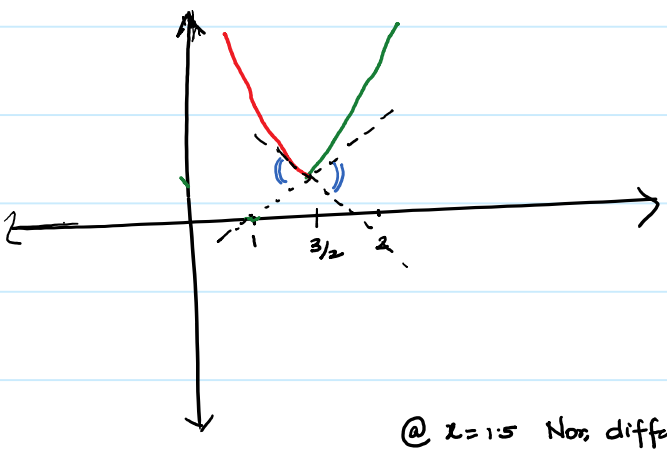
Sub gradients range :

$\nabla f(x) = 2x-2 = 2(-2)-2 = \boxed{-6}$

$[-2, 1]$ Any subgradient from the slope range can be considered

[for $x=-2$, $6x+y+4=0$]

(b) $g(x) = \max\{(x-1)^2, (x-2)^2\}$ @ $x=1.5, x=0$



$(x-1)^2 = (x-2)^2$

$x^2-2x+1 = x^2-4x+4$

$2x = 3, \quad x = 3/2$

@ $x=1.5$ Non differentiable

@ $x=0$ Differentiable

Sub gradients range :

$2(x-2), \quad 2(x-1)$

$[-1, 1]$

$2(x-2) = \boxed{-4}$

At $x=0$, $4x+y-4=0$

PROBLEM 1 : PERCEPTRON LEARNING :

1) 2) } Results and code are part of pdf **PERCEPTRON.pdf**

3) Here the rate of convergence doesnot change with step size

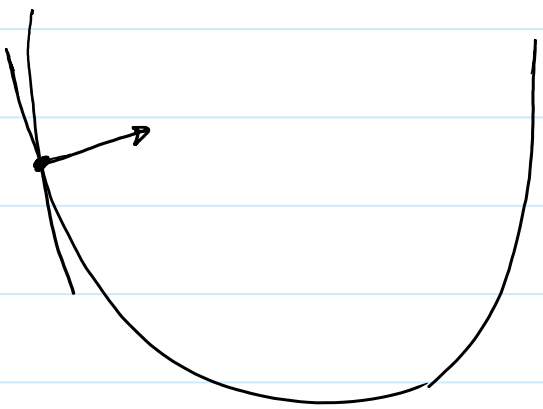
As, we are starting from $W = [0, 0, 0, 0]$ & $b = 0$

During the first step : Weight update : $\eta_t \nabla$

So, for different step sizes

after first step, all of them reach to the point on a plane that is

If we consider in (3D) indicated by the same unit normal vector



After first step, for different values they reach to the

same point on the optimization curve

Thus same no. of iterations happen for different step sizes

d)

Simplest dataset which is not linearly separable can be the case here.

So a single point with both the labels assigned to it can be considered or (1,1,+), (1,1,-) or

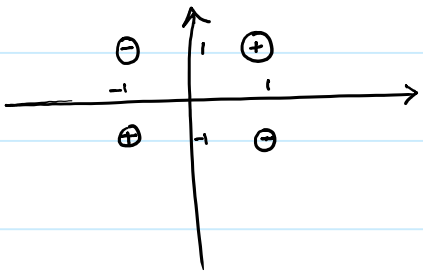
If the data is non ambiguous then collinear points with alternate class labels (+,-,+) can be considered as simple data set that fails to converge

So, for any data set that is not linearly separable can cause the algorithm to fail to converge.

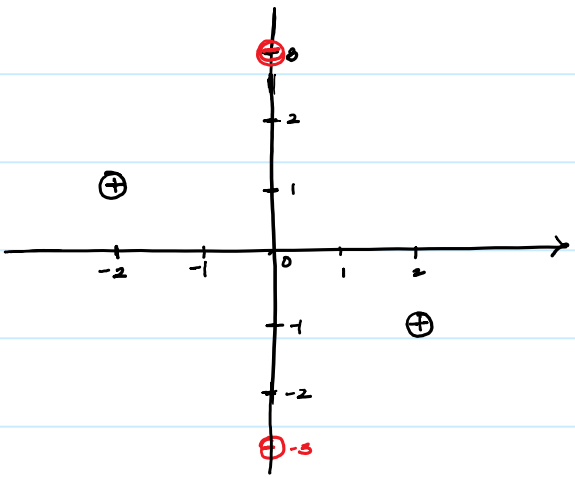
Other popular examples in 2D data set are data points resembling XOR.

II

1)



a) $\phi(x_1, x_2) = \begin{bmatrix} x_1 + x_2 \\ x_1 - 2x_1 \end{bmatrix} \rightarrow$



⊕ $(-1,-1) \rightarrow (-2,1)$ $\quad (1,1) \rightarrow (2,-1)$ ⊕

⊖ $(-1,1) \rightarrow (0,-3)$ $\quad (1,-1) \rightarrow (0,3)$ ⊖

Dataset is still not linearly separable

Proof Let $w_1 x_1 + w_2 x_2 + b$ be the line that separates data points [There doesnot exist a line]

$\Rightarrow$ $-2w_1 + w_2 + b > 0$, $2w_1 - w_2 + b > 0$ $\quad -3w_2 + b < 0$, $3w_2 + b < 0$

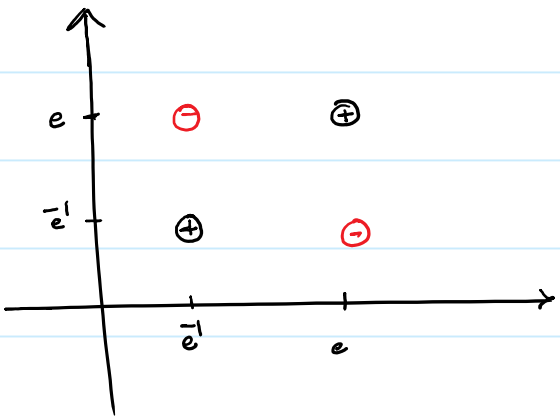
(Add) $b > 0$ $\quad$ (Add) $b < 0$

b)

$$\phi(x_1, x_2) = \begin{bmatrix} \exp(x_1) \\ \exp(x_2) \end{bmatrix}$$

$(-1,-1) \rightarrow (\bar{e}^1, \bar{e}^1)$ $\quad (1,1) \rightarrow (e,e)$

$(-1,1) \rightarrow (\bar{e}^1, e)$ $\quad (1,-1) \rightarrow (e, \bar{e}^1)$



Dataset is still not linearly separable

Proof Since applies as both in a way resemble XOR dataset

(Linear transformations of XOR dataset)

c) $\phi(x_1, x_2) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}$ ← (Non linear transformation)

$(-1,-1) \rightarrow (1,1,1)$ ⊕ $(1,1) \rightarrow (1,1,1)$ ⊕

$(-1,1) \rightarrow (1,1,-1)$ ⊖ $(1,-1) \rightarrow (1,1,-1)$ ⊖

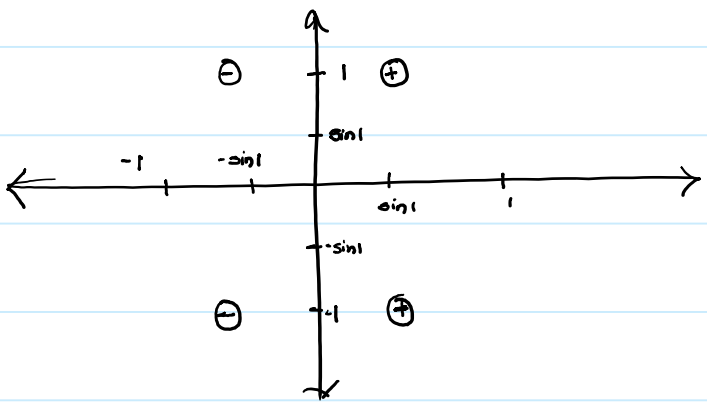
Points: $(1,1,1) \rightarrow$ ⊕ $(1,1,-1) \rightarrow$ ⊖

Linearly separable by a hyper plane ($z=0$)

d) $\phi(x_1, x_2) = \begin{bmatrix} x_1 \sin(x_2) \\ x_1 \end{bmatrix}$

$(-1,-1) \rightarrow (\sin 1, -1)$ ⊕ $(1,1) \rightarrow (\sin 1, 1)$ ⊕

$(-1,1) \rightarrow (-\sin 1, -1)$ ⊖ $(1,-1) \rightarrow (-\sin 1, 1)$ ⊖



Linearly separable by line $x=0$ (Non linear transformation)

2) For 2D data using gradient descent

Fit polynomial of degree k to data → Explain using feature vectors.

Per iteration complexity as function of size of feature representation &

No. of training data points.

$$f(x) = a_k x^k + a_{k-1} x^{k-1} + \cdots + b \quad \text{where } (a_k \neq 0)$$

↳ This can be represented features of size $[k]$

$$\Rightarrow X = [x^k \; x^{k-1} \; x^{k-2} \; \ldots \; x], \; W^T = [a_k \; a_{k-1} \; \ldots \; a_1]$$

$$\text{Squared error } (W, b) = \sum_{i=1}^{m} (y^i - a_k x^{ik} - a_{k-1} x^{i^{k-1}} \cdots a_1 x^i - b)^2$$

$$\frac{\partial E}{\partial a_j} = \sum_{i=1}^{m} 2(y^i - f(x^i)) \; (-x^{ij}) \quad \forall \; j \in [1, k]$$

$$\frac{\partial E}{\partial b} = \sum_{i=1}^{m} 2(y^i - f(x^i)) \; (-1)$$

$$\Rightarrow a_j = a_j + \gamma_b \frac{\partial E}{\partial a_j} \; \forall \; j \in [1, k] \quad \& \quad b = b + \gamma_t \frac{\partial E}{\partial b}$$

Complexity per iteration : $O(mk)$ ($m \rightarrow$ No. of data points)

($k \rightarrow$ size of feature vector)

3) $f(x) = \exp(ax+b)$

to a collection of data points $(x^{(1)}, y^{(1)}) \ldots (x^{(m)}, y^{(m)})$ in $\mathbb{R}^2$, $a, b \in \mathbb{R}$

1) Regression problem: (Minimizing squared error)

$$\text{Squared error} = \sum_{i=1}^{m} \left(y^{(i)} - f(x^{(i)}) \right)^2$$

$$= \sum_{i=1}^{m} \left(y^{(i)} - \exp(a x^{(i)} + b) \right)^2$$

2) Gradient descent to find minimum squared error

$$\text{Error}(a,b) = \sum_{i=1}^{m} \left(y^i - e^{ax^i+b} \right)^2$$

$$\frac{\partial E}{\partial a} = \sum_{i=1}^{m} 2 \left(y^i - e^{ax^i+b} \right) \left(-e^{ax^i+b} \right) x^i$$

$$\frac{\partial E}{\partial b} = \sum_{i=1}^{m} 2 \left(y^i - e^{ax^i+b} \right) \left(-e^{ax^i+b} \right)$$

$$\Rightarrow a_{new} = a_{old} - \gamma \frac{\partial E}{\partial a}, \quad b_{new} = b_{old} - \gamma \frac{\partial E}{\partial b}$$

$$\text{Error}(a,b) = \sum_{i=1}^{m} \left(y^{i2} - 2y^i e^{ax^i+b} + e^{2ax^i+2b} \right) = \sum_{i=1}^{m} y^{i2} - 2 \sum_{i=1}^{m} y^i e^{ax^i+b} + \sum_{i=1}^{m} e^{2ax^i+2b}$$

9) $Error(a,b) = \sum_{i=1}^{m} \left(y^{i^2} - 2y^i e^{ax^i+b} + e^{2ax^i+2b} \right) = \sum_{i=1}^{m} y^{i^2} - 2\sum_{i=1}^{m} y^i e^{ax^i+b} + \sum_{i=1}^{m} e^{2ax^i+2b}$

As we know that e^{ax} is convex, sum of all convex functions is convex

& product of a constant with convex function is convex

$\Rightarrow$ $\underbrace{\sum_{i=1}^{m} y^{i^2}}_{\text{constant}} - 2\underbrace{\sum_{i=1}^{m} y^i e^{ax^i+b}}_{\text{sum of convex functions}} + \underbrace{\sum_{i=1}^{m} e^{2ax^i+2b}}_{\text{sum of convex functions}}$

$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{Convex}}$

[$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$ [from taylor series expansion]

It is convex]

4. SUPPORT VECTOR MACHINES

I solved the primal problem for support vector machines
Here the data was not linearly separable, So I applied feature transformation and found corresponding weight vectors using quadprog solver of python.

Following is the feature transformation I have used.

$X : [x_1 \ x_2 \ x_3 \ x_4 \ 1]$

$\phi(x) : [x_1^2 \ x_2^2 \ x_3^2 \ x_4^2 \ x_1x_2 \ x_2x_3 \ x_3x_4 \ x_1x_3 \ x_1x_4 \ x_2x_4 \ x_1 \ x_2 \ x_3 \ x_4 \ 1]$

is the feature transformation used

I have attached results in a pdf file named SVM.pdf

showing code and results