ECE 544- Homework 2

Ahmadreza Eslaminia (Ae15)

Questions:

Ahmadreza Eslaminia ECE 544 HW2 ✻

AE15

1. [Max_Margin SVM] $D=\left\{\begin{bmatrix}x_1\\x_2\end{bmatrix}\right\}\in R^2$ $y\in\{-1,1\}$

$\min_{w,b}\frac{C}{2}\|w\|_2^2$ $\forall(x^{(i)},y^{(i)})\in D$ $y^{(i)}(w^Tx^{(i)}+b)\geq 1$

a). we will have 4 constraints as follows:

$x^1=\begin{bmatrix}1\\0\end{bmatrix}, y^1=1 \Rightarrow y^1(w^Tx^1+b)\geq 1 \Rightarrow w_1+b\geq 1$

$x^2=\begin{bmatrix}0\\1\end{bmatrix}, y^2=1 \Rightarrow y^2(w^Tx^2+b)\geq 1 \Rightarrow w_2+b\geq 1$

$w=\begin{bmatrix}w_1\\w_2\end{bmatrix}$

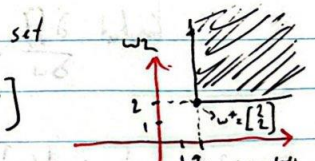
$x^3=\begin{bmatrix}0\\0\end{bmatrix}, y^3=-1 \Rightarrow y^3(w^Tx^3+b)\geq 1 \Rightarrow -b\geq 1$

$x^4=\begin{bmatrix}-1\\-1\end{bmatrix}, y^4=-1 \Rightarrow y^4(w^Tx^4+b)\geq 1 \Rightarrow -(-w_1-w_2+b)\geq 1$

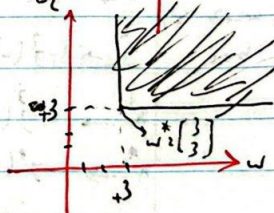
$\begin{cases}w_1+b\geq 1\\ w_2+b\geq 1\\ b\leq -1\\ w_1+w_2\geq 1+b\end{cases}$

b) $b=0$ $\Rightarrow$ does not obey $b\leq -1$ condition $\Rightarrow$ no feasible set

$b=-1:\begin{cases}w_1+b\geq 1 \Rightarrow w_1\geq 2\\ w_2+b\geq 1 \Rightarrow w_2\geq 2\\ b=-1\ \checkmark\end{cases}$ $\Rightarrow$ optimum $w^*=\begin{bmatrix}+2\\+2\end{bmatrix}$



$b=-2:\begin{cases}w_1+b\geq 1 \Rightarrow w_1\geq 3\\ w_2+b\geq 1 \Rightarrow w_2\geq 3\\ b=-2\ \checkmark\end{cases}$ $\Rightarrow$ optimal $w^*=\begin{bmatrix}+3\\+3\end{bmatrix}$



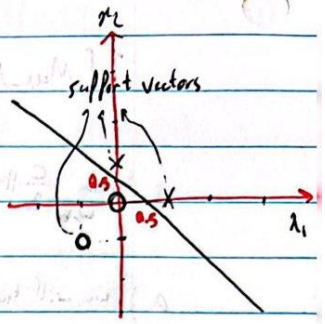
Since $b\geq -1 \Rightarrow \begin{cases}w_1\geq 1-b\\ w_2\geq 1-b\end{cases} \Rightarrow \begin{cases}w_1\geq 2\\ w_2\geq 2\end{cases}$ $\Rightarrow$ For this objective function $w^*=\begin{bmatrix}2\\2\end{bmatrix}$ is the optimal choice on feasible space with $b=-1$

P4

7.c)

we find points $x^1=\begin{bmatrix}1\\0\end{bmatrix}$, $x^2=\begin{bmatrix}0\\1\end{bmatrix}$, $x^3=\begin{bmatrix}0\\0\end{bmatrix}$

are support vector points.



$$\Rightarrow \begin{cases} x^1 \Rightarrow w_1+b=1 \\ x^2 \Rightarrow w_2+b=1 \\ x^3 \Rightarrow b=-1 \end{cases} \Rightarrow \begin{cases} b=-1 \\ w_1=w_2=2 \end{cases} \Rightarrow w_1x_1+w_2x_2+b=0$$

$$= 2x_1+2x_2=1$$

d) they are linearly seperable s.t. Eq.7 has a feasible solution.

e) $\min \frac{c}{2}\|w\|_2^2 + \sum_{i=0} \xi^i \quad$ s.t $\xi_i = \max\{0,\ 1-y^{(i)}(w^T x^{(i)}+b)\}$

$$\Rightarrow \min \frac{c}{2}\|w\|_2^2 + \sum_{i=0}^{\max}\{0,\ 1-y^{(i)}(w^T x^{(i)}+b)\}$$

Gradient: $\frac{\partial\{\}}{\partial w} = \left[c\begin{bmatrix}w_1\\w_2\end{bmatrix} + \sum_{i=0} -y^{(i)}\begin{bmatrix}x_1^{(i)}\\x_2^{(i)}\end{bmatrix} \right]$ For $x^i \neq 0$ that $1-y^{(i)}(w^T x^{(i)}+b)>0$

for $w_1=2$, $w_2=2$, $b=-1$:

$x^1 \rightarrow 1-y^1(w^T x^1+b) = 1-(2-1) = 0$

$x^2 \rightarrow 1-y^2(w^T x^2+b) = 1-(2-1) = 0$

$x^3 \rightarrow 1-y^3(w^T x^3+b) = 1-1 = 0$

$x^4 \rightarrow 1-y^4(w^T x^4+b) = 1+(-4-1) = -4$

$$\Rightarrow \frac{\partial\{\}}{\partial w} = c\begin{bmatrix}2\\2\end{bmatrix} = \begin{bmatrix}2c\\2c\end{bmatrix}$$

Since $c>0 \Rightarrow \frac{\partial\{\}}{\partial w} \neq 0$ so it is not optimal

solution driven from the new function.

7.f) from previous section we found the $w=\begin{bmatrix}2\\2\end{bmatrix}$ is not optimal.

From the python file the solution is $w^*=\begin{bmatrix}0.667\\0.667\end{bmatrix}$ $b=0.333$ loss: 1.778835

we can see by introducing new terms to loss the solution mis classifies one point ($c=7$)

however, by choosing c closer to zero we can give more importance to the margin terms & the problem can be solved.

2. [L2 SVM] $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{101}$ $x^{(i)} \in \mathbb{R}^d$ $y^{(i)} \in \{-1, 1\}$

$$\min \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{101} (\xi^{(i)})^2 \quad \text{s.t. } y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi^{(i)} \quad i \in \{1, \dots, 101\}$$

$\xi^{(i)} \geq 0$ ↓ #

a) $\frac{1}{2}(w^*)^T w^* + \frac{C}{2}((\xi^{(1)})^*)^2 + \frac{C}{2}((\xi^{(2)})^*)^2$

b) since $(\xi^{(3)})^*$ is not zero $(\xi^{(3)})^* \neq 0 \Rightarrow$ loss function is as follow:

$\frac{1}{2}(w^*)^T w^* + \frac{C}{2}((\xi^{(1)})^*)^2 + \frac{C}{2}((\xi^{(2)})^*)^2 + \frac{C}{2}((\xi^{(3)})^*)^2 > \frac{C}{2}$ previous loss

c) we show that $\tilde{\xi}^{(i)} = 0$ does not violate the previous constraint so it is feasible solution

Priously we had $(\xi^{(i)})^* < 0$ which results in constrain below:

$$y^{(i)}((w^*)^T x^{(i)} + b^*) \geq 1 - (\xi^{(i)})^* > 1 \quad \text{(I)}$$

however if we consider $\tilde{\xi}^{(i)} = 0$ we have the following constraint:

$$y^{(i)}((w^*)^T x^{(i)} + b^*) \geq 1 \quad \text{(II)}$$

As you can see (II) does not violate (I) which means $\tilde{\xi}^{(i)} = 0$ is a feasible solution

d) As we have shown in previous sections not only $(\hat{w}, \hat{b}, \hat{\xi})$ is a feasible solution but also it results in smaller loss. so this would violate the first assumption that (w^*, b^*, ξ^*) is optimal solution. So we can conclude that for the optimal solution $\xi^{(i)} \geq 0$ always hold so we can remove it.

P3

2.e) The lagrangian would be as follows: 5/6

$$L(w,b,\xi,\alpha,\gamma,x^{(i)},c,D) = \frac{1}{2}w^Tw + \frac{c}{2}\sum_{i=1}^{|D|}(\xi^{(i)})^2 - \sum_{i=1}^{D}\alpha^{(i)}\left(y^{(i)}(w^Tx^{(i)}+b) - 1 + \xi^{(i)}\right)$$

$\forall i \Rightarrow \alpha^{(i)} \geq 0$

f) show the dual of problem is $\max_{\alpha \geq 0} -\frac{1}{2}\alpha^T(Q + I\frac{1}{c})\alpha + 1^T\alpha$

s.t. $y^T\alpha = 0$

where $Q_{ij} = y_iy_j x_i^T x_j$ $\alpha \in R^{|D|}$ $y \in R^{|D|}$

Take following partial derivatives:

$$\nabla_w L = 0 \Rightarrow w - \sum_{i=1}^{|D|}\alpha^{(i)}y^{(i)}x^{(i)} = 0 \Rightarrow w = \sum_{i=1}^{|D|}\alpha^{(i)}y^{(i)}x^{(i)}$$

$$\nabla_b L = 0 \Rightarrow \sum_{i=1}^{|D|}\alpha^{(i)}y^{(i)} = 0$$

$$\nabla_{\xi^{(i)}} L = 0 \Rightarrow c\xi^{(i)} - \alpha^{(i)} = 0 \Rightarrow \xi^{(i)} = \frac{\alpha^{(i)}}{c}$$ / placing above in Lagrangian:

$$L = \frac{1}{2}\left(\sum_{i=1}^{|D|}(\alpha^i y^{(i)} x^{(i)})\right)^T\left(\sum_{j=1}^{|D|}\alpha^j y^j x^j\right) + \frac{c}{2}\sum_{i=1}^{|D|}\frac{(\alpha^i)^2}{c^2} -$$

$$-\sum_{i=1}^{|D|}\alpha^{(i)}y^{(i)}\left(\sum_{j=1}^{|D|}x^Ty^jT\alpha^j\right)x^{(i)} + b\Big| + \sum_{i=1}^{|D|}\alpha_i - \sum_{i=1}^{|D|}\alpha_i\xi_i$$

$$= -\frac{1}{2}\alpha^T\left(\sum_{i=1}^{|D|}\sum_{j=1}^{|D|}y_iy_jx_i^Tx_j\right)\alpha - \frac{1}{2}\alpha^T(I\frac{1}{c})\alpha + 1^T\alpha = -\frac{1}{2}\alpha^T(Q+I\frac{1}{c})\alpha + 1^T\alpha$$

$\sum_{i=1}^{|D|}\alpha^{(i)}y^{(i)} = 0 \Rightarrow y^T\alpha = 0$ ✓

3. [Support Vector Machine] $\min \frac{1}{2}\|w\|_2^2$ s.t $y^{(i)}(w^T x^{(i)} + b) \geq 1 \quad \forall (x^{(i)}, y^{(i)}) \in D$

a) $\Rightarrow$ lagrangian with multipliers α_i :

$$L(w, b, \alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{|D|} \alpha^{(i)}\left(1 - y^{(i)} w^T x^{(i)} + b\right)$$

b) consider $L(w, b, \alpha) = \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{|D|} \alpha^{(i)}\left(1 - y^{(i)} w^T x^{(i)}\right)$

$$\nabla_w L = 0 \Rightarrow w - \sum_{i=1}^{|D|} \alpha^{(i)} y^{(i)} x^{(i)} = 0 \Rightarrow \underline{w = \sum_{i=1}^{|D|} \alpha^i y^{(i)} x^{(i)}}$$

$$\Rightarrow \text{let } L = \frac{1}{2}\left\| \sum_{i=1}^{|D|} \alpha^i y^{(i)} x^{(i)} \right\|_2^2 + \sum_{i=1}^{|D|} \alpha^i - \underbrace{\sum_{i=1}^{|D|} \alpha^i y^i \left(\sum_{j=1}^{|D|} \alpha^j y^{(j)} x^{(j)} \right) x^i}_{= \left\| \sum \alpha^i y^i x^i \right\|_2^2}$$

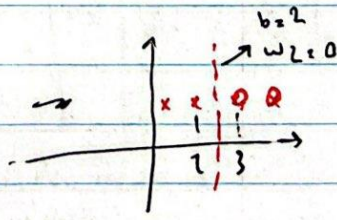
$$= \sum_{i=1}^{|D|} \alpha^i - \frac{1}{2}\left\| \sum_{i=1}^{|D|} \alpha^{(i)} y^{(i)} x^{(i)} \right\|_2^2$$

s.t $\alpha_i \geq 0 \quad \forall i \in D$

c) $k(x, z) = (x^T z)^2 + 1 = \phi(x)^T \phi(z) \xrightarrow{n \text{ is 2Dim}} \phi(x) = \begin{bmatrix} x^2 \\ 1 \end{bmatrix}$

	i	$x^{(i)}$	$\phi(x_i)$	y_i
	1	$\frac{1}{2}$	$\begin{bmatrix} \frac{1}{4} \\ 1 \end{bmatrix}$	+1
$\Rightarrow$	2	-1	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	+1
	3	$\sqrt{3}$	$\begin{bmatrix} 3 \\ 1 \end{bmatrix}$	-1
	4		$\begin{bmatrix} 16 \\ 1 \end{bmatrix}$	-1

2, 3 support vectors $\longrightarrow$ mid point is $\frac{1}{2}$



$\Rightarrow$ $w^T = [w^1, w^2]$

$$\Rightarrow \boxed{w_1 = -1, \; b = 2}$$

$p_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow w^T p + b = 1$

$p_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \Rightarrow w^T p + b = 0$

$p_3 = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \Rightarrow w^T p + b = -1$

$P_5 =$

3.d) points of ②, ③ are support vectors.

we have: $w^* = \sum_{i=1}^{N} \alpha^{(i)} y^{(i)} z^{(i)}$ $\quad z = \phi(x)$

$\alpha^{(1)} = \alpha^{(4)} = 0$

$\Rightarrow \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \alpha_2 \times 1 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \alpha_3 \times -1 \times \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ $\Rightarrow \alpha_{(2)} = \alpha_{(3)} = 1/2$

$\alpha^{(1)} = \alpha^{(4)} = 0$ (not support vector)

4. [Multi logistic regression] $\min -\sum_{(x,y)\in D} \log P(y|x)$ where $P(y|x) = \frac{e^{w_y^T \begin{bmatrix} x \\ 1 \end{bmatrix}}}{\sum_{j\in\{0,1,2\}} e^{w_j^T \begin{bmatrix} x \\ 1 \end{bmatrix}}}$

a) $w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$ where $w_0 = \begin{bmatrix} w_0^{(2)} \\ w_0^{(1)} \\ w_0^{(b)} \end{bmatrix}$ $w_1 = \begin{bmatrix} w_1^{(2)} \\ w_1^{(1)} \\ w_1^{(b)} \end{bmatrix}$ $w_2 = \begin{bmatrix} w_2^{(2)} \\ w_2^{(1)} \\ w_2^{(b)} \end{bmatrix}$

(2 weight + 1 bais)

we have one w_y for each classes. so since we have 3 parameter at each class and 3 classes $\Rightarrow 3\times 3 = 9$ Parameter $\Rightarrow$ dim of $w \Rightarrow 9\times 1$

b) $w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$ $\psi(x,y) = \begin{bmatrix} \phi(x) & \delta(y=0) \\ \phi(x) & \delta(y=1) \\ \phi(x) & \delta(y=2) \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} & \delta(y=0) \\ \begin{bmatrix} x \\ 1 \end{bmatrix} & \delta(y=1) \\ \begin{bmatrix} x \\ 1 \end{bmatrix} & \delta(y=2) \end{bmatrix}$

$w^T \psi(x,y) = w_0^T \begin{bmatrix} x^{(1)} \\ 1 \end{bmatrix} \delta(y=0) + w_1^T \begin{bmatrix} x^1 \\ 1 \end{bmatrix} \delta(y=1) + w_2^T \begin{bmatrix} x^2 \\ 1 \end{bmatrix} \delta(y=2)$

$= w_{y^{(i)}}^T \begin{bmatrix} x^{(1)} \\ 1 \end{bmatrix}$ $\forall y^{(i)}$ as $i \in \{0,1,2\}$.

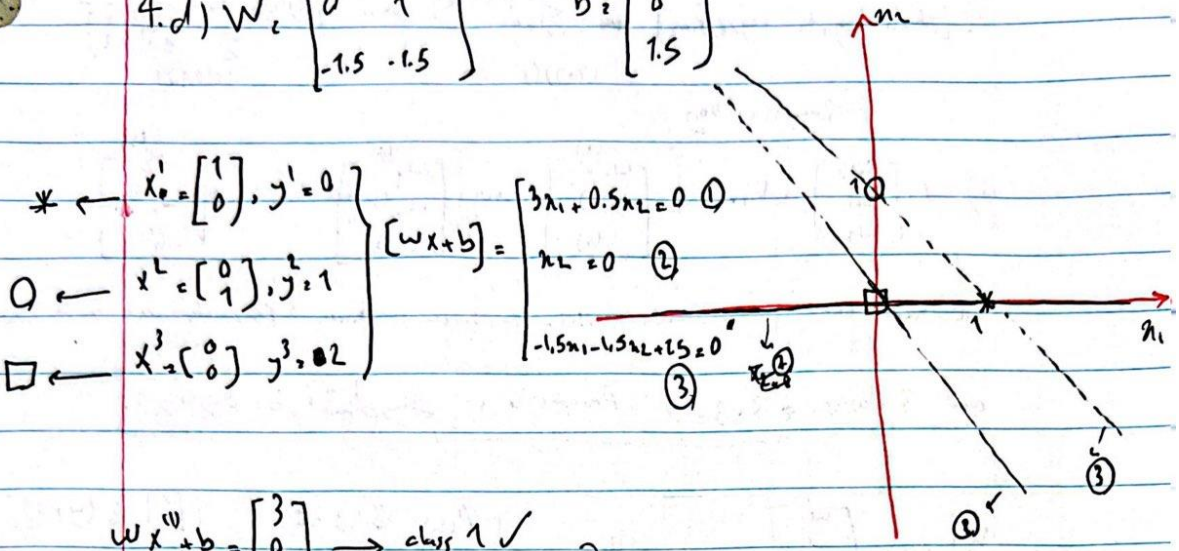
c) we had F as following in past:

$F(w,x,y) = \begin{bmatrix} w_0^T \begin{bmatrix} x \\ 1 \end{bmatrix} \\ w_1^T \begin{bmatrix} x \\ 1 \end{bmatrix} \\ w_2^T \begin{bmatrix} x \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} w_0^{(2)} x_2 + w_0^{(1)} x_1 + b_0 \\ w_1^{(2)} x_2 + w_1^{(1)} x_1 + b_1 \\ w_2^{(2)} x_2 + w_2^{(1)} x_1 + b_2 \end{bmatrix} = \begin{bmatrix} W x + b \end{bmatrix}$

where $W = \begin{bmatrix} w_0^{(2)} & w_0^{(1)} \\ w_1^{(2)} & w_1^{(1)} \\ w_2^{(2)} & w_2^{(1)} \end{bmatrix}_{3\times 2}$ $b = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}_{3\times 1}$

in this case row of W are the w vector with ignoring biases. That's how it relate to previous w.

4.d) $W = \begin{bmatrix} 3 & 0.5 \\ 0 & 1 \\ -1.5 & -1.5 \end{bmatrix}$ $b = \begin{bmatrix} 0 \\ 0 \\ 1.5 \end{bmatrix}$



* ← $x^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $y^1 = 0$

○ ← $x^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $y^2 = 1$

□ ← $x^3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $y^3 = 2$

$[wx+b] = \begin{cases} 3x_1 + 0.5x_2 = 0 & (1) \\ x_2 = 0 & (2) \\ -1.5x_1 - 1.5x_2 + 1.5 = 0 & (3) \end{cases}$

$wx^{(1)} + b = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}$ → class 1 ✓

$wx^{(2)} + b = \begin{bmatrix} 0.5 \\ 1 \\ 0 \end{bmatrix}$ → class 2 ✓

$wx^{(3)} + b = \begin{bmatrix} 0 \\ 0 \\ 1.5 \end{bmatrix}$ → class 3 ✓

As all numbers are positive we can see the higher magnitud will show the class because e^n is not going to change that.

So the model correctly classifies all the cluster points

e) $W = \begin{bmatrix} 8.7387, & -1.6501 \\ -1.9086, & 9.0514 \\ -7.2665, & -6.9575 \end{bmatrix}$ $b = \begin{bmatrix} -2.5121 \\ -2.5161 \\ 5.3407 \end{bmatrix}$

$P(y|x) = \begin{bmatrix} 9.9981 \times 10^{-1}, & 1.7380 \times 10^{-5}, & 6.9603 \times 10^{-4} \\ 3.0772 \times 10^{-5}, & 9.9987 \times 10^{-1}, & 9.4992 \times 10^{-9} \\ 1.6024 \times 10^{-9}, & 2.1721 \times 10^{-9}, & 9.9835 \times 10^{-1} \end{bmatrix}$

P8

5. [Multi class classification via NN]

a) Since it has to be at least one class $\sum F(z) \geq 1$ => Softmax and $0 \leq f(z_i) \leq 1$

$$f(z_i) = \frac{e^{z_i}}{\sum_{i=1}^{3} e^{z_i}}$$

b) $G(z) = \left(\frac{z_1}{z_1+z_2+z_3}, \frac{z_2}{*}, \frac{z_3}{*} \right)$

Since f uses exponential function we know that f is translation invariant.

about G => $G(z^{(1)}) = \left(\frac{0.1}{0.4}, \frac{0.1}{0.4}, \frac{0.2}{0.2} \right) = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{2} \right)$

$G(z^{(1)}) = \left(\frac{1.01}{3.04}, \frac{1.01}{3.04}, \frac{1.04}{3.04} \right) = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right)$

} G is not translation invariant

we can understan as translation invariant function (such as f) is more desirable than a non * * Function (such as G) since the non translation one if the numbers get large in comparison to their difference they are going to give same output for all the inputs.

c) $CE(y, f(z)) = -\ln \left(\frac{e^{z_3}}{e^{z_1}+e^{z_2}+e^{z_3}} \right) = -z_3 + \ln(e^{z_1}+e^{z_2}+e^{z_3})$

$$\frac{\partial CE(y, f(z))}{\partial z_3} = -1 + \frac{e^{z_3}}{e^{z_1}+e^{z_2}+e^{z_3}}$$

$CE(y, G(z)) = -\ln \left(\frac{z_3}{z_1+z_2+z_3} \right) = -\ln^{z_3} + \ln^{z_1+z_2+z_3}$

$$\frac{\partial CE(y, G(z))}{\partial z_3} = -\frac{1}{z^3} + \frac{1}{z_1+z_2+z_3}$$

} As you can see At $z=(1, 10^{-5}, 10^{-5})$ gradient of $CE(y, G(z))$ will explode and will result to optimization issue.

5.d) one-vs-rest is going to be slower. Since in Multin class classification Although the last layer has more parameters (3x), the whole computational cost (forward) for 3 of one-vs-rest NN which is needed to correctly assign a correct class to input is higher than multi class classification.

Codes A3-SVM:

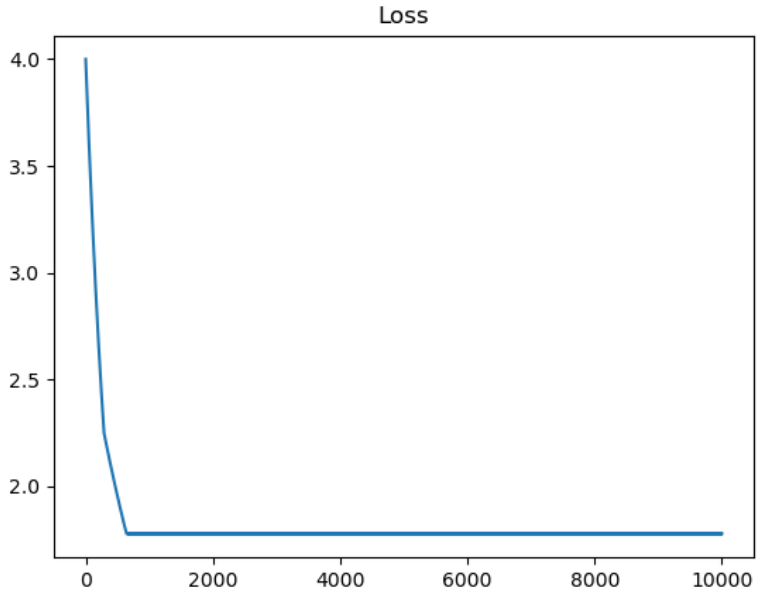
C: > DriveA > UIUCcourses > Fall 2023 > ECE 544 pattern recognition > HWS > hw2 > homework2 > A3_SVM.

```python
import torch
import torch.optim as optim
import torch.nn as nn
import matplotlib.pyplot as plt

torch.manual_seed(1)
X = torch.tensor([[1, 0, 0, -1],[0, 1, 0, -1]], dtype=torch.float32)
y = torch.tensor([1, 1, -1, -1], dtype=torch.float32)
# initialize w and b
w = torch.tensor([0.0, 0.0], requires_grad=True)
b = torch.tensor([0.0], requires_grad=True)
alpha = 0.001
C = 1

optimizer = optim.SGD([w,b], lr=alpha, weight_decay=0)
optimizer.zero_grad()

grads = []
losses = []

for iter in range(10000):
    if iter==0:
        print('1-y(w^T*x+b): {}'.format(1 - y*(torch.matmul(X.T, w)+b)))
    ###############################
    ## Complete this line which is our cost function
    ## Dimensions: loss (scalar)
    ###############################
```

```
    ## Complete this line which is our cost function
    ## Dimensions: loss (scalar)
    ############################

    New_term = 1 - y*(torch.matmul(X.T, w)+b)
    loss = C/2 * torch.matmul(w, w) + torch.sum(torch.clamp(New_term, min=0))
    loss.backward()
    gn = torch.norm(w.grad)**2 + torch.norm(b.grad)**2
    print("Iter: %d; Loss: %f; ||Grad||: %f" % (iter, loss, gn))
    optimizer.step()
    optimizer.zero_grad()

    losses.append(loss.item())
    grads.append(gn)

    print('w: {}'.format(w.data))
    print('b: {}'.format(b.data))

plt.figure()
plt.plot(losses, label='Loss')
plt.title('Loss')
plt.show()
plt.figure()
plt.plot(grads, label='Grad')
plt.title('||Grad||')
plt.show()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Iter: 9997; Loss: 1.778837; ||Grad||: 1.891705
w: tensor([0.6671, 0.6671])
b: tensor([0.3320])
Iter: 9998; Loss: 1.778868; ||Grad||: 1.221706
w: tensor([0.6674, 0.6674])
b: tensor([0.3330])
Iter: 9999; Loss: 1.778391; ||Grad||: 1.890811
w: tensor([0.6667, 0.6667])
b: tensor([0.3320])
```


Loss
4.0
3.5
3.0
2.5
2.0
0
2000
4000
6000
8000
10000

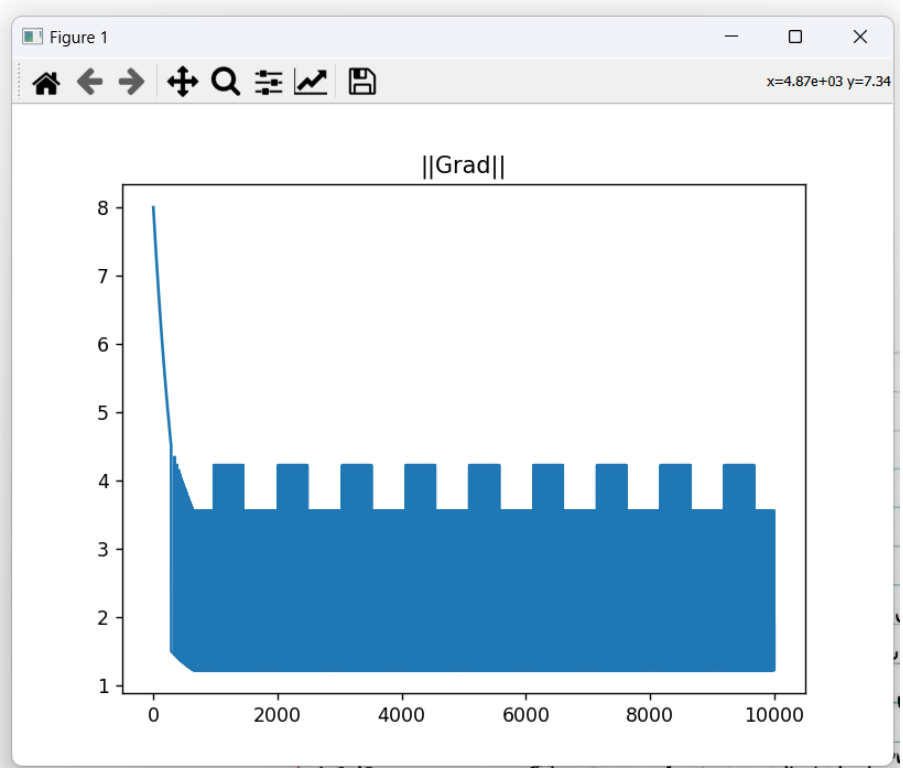


Figure 1
x=4.87e+03 y=7.34
||Grad||
8
7
6
5
4
3
2
1
0
2000
4000
6000
8000
10000

A4-Multiclass:

```
import torch
import torch.optim as optim
import torch.nn as nn

torch.manual_seed(1)
alpha = 1
C = 0

##############################
## encode the dataset to fit the one specified in HW4.pdf (note that bias
## is part of the network now)
## Dimensions: X (2x3); y (3)
##############################
X = torch.Tensor([[1, 0, 0],[0, 1, 0]])
y = torch.LongTensor([0, 1, 2])


class ShallowNet(nn.Module):
    def __init__(self):
        super(ShallowNet, self).__init__()
        self.fc1 = nn.Linear(2,3, bias=True)

    def forward(self, X):
        return self.fc1(X)

net = ShallowNet()
print(net)

print(net(torch.transpose(X,0,1)).squeeze())

optimizer = optim.SGD(net.parameters(), lr=alpha, weight_decay=C)
optimizer.zero_grad()

criterion = nn.CrossEntropyLoss()

for iter in range(10000):
    netOutput = net(torch.transpose(X,0,1))
```

C: > DriveA > UIUCcourses > Fall 2023 > ECE 544 pattern recognition > HWS > hw2 > homework2 > A4_Multiclass.py

```
31    optimizer = optim.SGD(net.parameters(), lr=alpha, weight_decay=C)
32    optimizer.zero_grad()
33
34    criterion = nn.CrossEntropyLoss()
35
36    for iter in range(10000):
37        netOutput = net(torch.transpose(X,0,1))
38
39        ##############################
40        ## provide the arguments for the criterion function
41        ## Dimensions: loss (scalar)
42        ##############################
43        loss = criterion( netOutput , y )
44
45        loss.backward()
46        gn = 0
47        for f in net.parameters():
48            gn = gn + torch.norm(f.grad)
49        print("Loss: %f; ||g||: %f" % (loss, gn))
50
51        ##############################
52        ## Use two functions within the optimizer instance to perform the update step
53        ##############################
54        optimizer.step()
55        optimizer.zero_grad()
56
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Parameter containing:
tensor([[ 8.7387, -1.6501],
        [-1.9086,  9.0514],
        [-7.2685, -6.9575]], requires_grad=True)
Parameter containing:
tensor([-2.5121, -2.5161,  5.3407], requires_grad=True)
tensor([[9.9981e-01, 1.7380e-05, 6.9603e-04],
        [3.0772e-05, 9.9987e-01, 9.4992e-04],
        [1.6024e-04, 1.1721e-04, 9.9835e-01]], grad_fn=<SoftmaxBackward0>)
PS C:\DriveA\UIUCcourses\Fall 2023\ECE 544 pattern recognition\HWS\hw2\homework2>
```