Saaif Ahmed Exam 2

Problem 1

A:

$$\nabla_\theta = \frac{1}{2}\sum_{i=1}^{N} 2\left(\vec{\theta}^T\overrightarrow{x_i} - y_i\right) * (\overrightarrow{x_i}) + \frac{\lambda}{2}2\left(\left||\vec{\theta}|\right|\right) * \frac{\vec{\theta}}{||\vec{\theta}||}$$

$$= \frac{1}{2}\sum_{i=1}^{N} 2\left(\vec{\theta}^T\overrightarrow{x_i} - y_i\right) * (\overrightarrow{x_i}) + \lambda\vec{\theta}$$

Thus at some iteration k

$$\vec{\theta}^k = \vec{\theta}^{(k-1)} - \alpha\left(\frac{1}{2}\sum_{i=1}^{N} 2\left(\vec{\theta}^T\overrightarrow{x_i} - y_i\right) * (\overrightarrow{x_i}) + \lambda\vec{\theta}\right)$$

B:

$$L(\vec{\theta}) = \frac{1}{2}\left(X\vec{\theta} - \vec{y}\right)^T\left(X\vec{\theta} - \vec{y}\right) + \frac{\lambda}{2}\left||\vec{\theta}|\right|^2$$

$$\nabla_{\vec{\theta}} = X^TX\vec{\theta} - X^T\vec{y} + \lambda\vec{\theta}$$

$$\vec{\theta}^* = (X^TX)^{-1}X^T\vec{y} + \lambda\vec{\theta}$$

$$X = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix} ; y = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$\vec{\theta}^* = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}^{-1} * \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}^T * \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}^T * \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ -1 \end{bmatrix}$$

$$\vec{\theta}^* = \begin{bmatrix} -2 \\ -1 \end{bmatrix}$$

Problem 1

C:

$$\vec{\theta}^{k} = \vec{\theta}^{(k-1)} - \alpha\left(\frac{1}{2}\sum_{i=1}^{N} 2\left(\vec{\theta}^{T}\vec{x_i} - y_i\right) * \left(\vec{x_i}\right) + \lambda\vec{\theta}\right)$$

$$\theta^1 = \begin{bmatrix}0\\0\end{bmatrix} - \left(\left(\begin{bmatrix}0\\0\end{bmatrix}^T\begin{bmatrix}0\\1\end{bmatrix} + 1\right) * \begin{bmatrix}0\\1\end{bmatrix} + \left(\begin{bmatrix}0\\0\end{bmatrix}^T\begin{bmatrix}-1\\1\end{bmatrix} - 1\right) * \begin{bmatrix}-1\\1\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix}\right)$$

$$\theta^1 = \begin{bmatrix}0\\0\end{bmatrix} - \left(\begin{bmatrix}0\\1\end{bmatrix} + \begin{bmatrix}1\\-1\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix}\right)$$

$$\theta^1 = \begin{bmatrix}-1\\0\end{bmatrix}$$

$$\theta^2 = \begin{bmatrix}-1\\0\end{bmatrix} - \left(\left(\begin{bmatrix}-1\\0\end{bmatrix}^T\begin{bmatrix}0\\1\end{bmatrix} + 1\right) * \begin{bmatrix}0\\1\end{bmatrix} + \left(\begin{bmatrix}-1\\0\end{bmatrix}^T\begin{bmatrix}-1\\1\end{bmatrix} - 1\right) * \begin{bmatrix}-1\\1\end{bmatrix} + \begin{bmatrix}-1\\0\end{bmatrix}\right)$$

$$= \begin{bmatrix}-1\\0\end{bmatrix} - \left(\begin{bmatrix}0\\1\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix} + \begin{bmatrix}-1\\0\end{bmatrix}\right)$$

$$= \begin{bmatrix}0\\-1\end{bmatrix}$$

D:

$$L\left(\vec{\theta}^2\right) = \frac{1}{2}\left(\left(\begin{bmatrix}0\\-1\end{bmatrix}^T\begin{bmatrix}0\\1\end{bmatrix} + 1\right)^2 + \left(\begin{bmatrix}0\\-1\end{bmatrix}^T\begin{bmatrix}-1\\1\end{bmatrix} - 1\right)^2\right) + \frac{1}{2}\left|\left|\begin{bmatrix}0\\-1\end{bmatrix}\right|\right|^2$$

$$= \frac{1}{2}(0 + 4) + \frac{1}{2}$$

$$= \frac{5}{2}$$

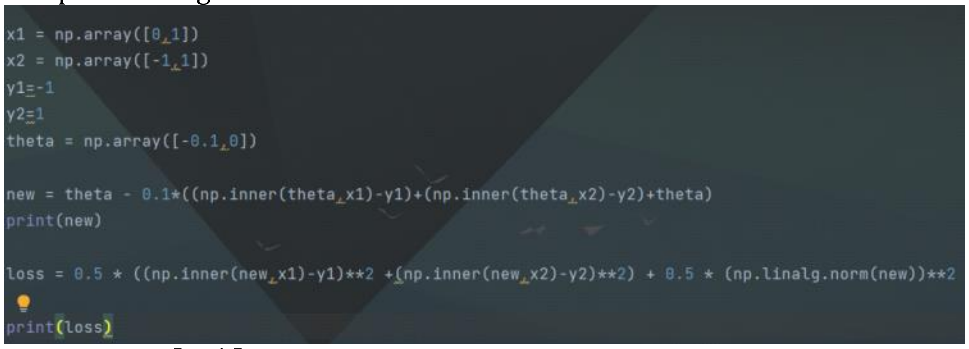
$$L\left(\vec{\theta}^0\right) = \frac{1}{2}\left(\left(\begin{bmatrix}0\\0\end{bmatrix}^T\begin{bmatrix}0\\1\end{bmatrix} + 1\right)^2 + \left(\begin{bmatrix}0\\0\end{bmatrix}^T\begin{bmatrix}-1\\1\end{bmatrix} - 1\right)^2\right)$$

$$= \frac{1}{2}(1 + 1)$$

$$= 1$$

In this case the loss of $L(\vec{\theta}^2) > L(\vec{\theta}^0)$. This is due to the learning rate being too large since $\alpha = 1$

With a smaller learning rate it the loss decreases. Such as $\alpha = 0.1$ which I have hard computed. using the code below



```
x1 = np.array([0,1])
x2 = np.array([-1,1])
y1=-1
y2=1
theta = np.array([-0.1,0])

new = theta - 0.1*((np.inner(theta,x1)-y1)+(np.inner(theta,x2)-y2)+theta)
print(new)

loss = 0.5 * ((np.inner(new,x1)-y1)**2 +(np.inner(new,x2)-y2)**2) + 0.5 * (np.linalg.norm(new))**2

print(loss)
```

$* \ Note: \vec{\theta}^1 = \begin{bmatrix}-\frac{1}{10}\\0\end{bmatrix}$ using the work from part c with $\alpha = 0.1$

Because the loss is not reduced after iterations it does not satisfy the theory of gradient descent

Problem 2:

A:

Proof of symmetry:

We have that $K(x, x') = (xx' + 1)^{2021}$

And we have that $K(x', x) = (x'x + 1)^{2021} = (xx' + 1)^{2021}$

Thus symmetric

Proof of Positive Semi Definite

Define $\overline{K} = (xx' + 1)$

Since $K = < \phi(x), \phi(x') >$

Where $\phi(x) = \begin{bmatrix} x \\ 1 \end{bmatrix}$; $\phi(x') = \begin{bmatrix} x' \\ 1 \end{bmatrix}$

We have that $\overline{K}$ is a valid kernel function by definition

We can build K^n by defining $K_{m+1} = K_m \odot \overline{K}$

We know that a kernel function created from the Hadamard product is a valid kernel function.

Thus by induction we can see that K_{m+1} is a valid kernel function

$K_1 = \overline{K}$ By induction we have that $K(x, x') = (xx' + 1)^{2021}$ is a valid kernel function.

B:

Proof of symmetry

$K(x, x') = f(x)K(x, x')f(x')$

Consider

$K(x, x') = f(x')K(x', x)f(x) = f(x)K(x, x')f(x')$

Since $f(x) = R \to R$

Thus symmetric

Proof of PSD

$f(\vec{x})$ is a vector of $f(x)$ being applied to each component in $\vec{x}: \vec{x} \in R^n$

Consider $\vec{y}^T K \vec{y}: \vec{y} \in R^n$

$\vec{y}^T K \vec{y} = Tr\left(diag(\vec{y}) * K * diag(\vec{y})\right)$

$K = diag\left(f(\vec{x})\right) * K_1 * diag\left(f(\vec{x}')\right)$

$\vec{y}^T K \vec{y} = Tr\left(diag(\vec{y}) * diag\left(f(\vec{x})\right) * K_1 * diag\left(f(\vec{x}')\right) * diag(\vec{y})\right)$

K is PSD so consider square roots

$Tr\left(diag(\vec{y}) * diag\left(f(\vec{x})\right) * K_1^{\frac{1}{2}} * K_1^{\frac{1}{2}} * diag\left(f(\vec{x}')\right) * diag(\vec{y})\right)$

$Tr\left(K_1^{\frac{1}{2}} * diag(\vec{y}) * diag\left(f(\vec{x})\right) * K_1^{\frac{1}{2}} * diag\left(f(\vec{x}')\right) * diag(\vec{y})\right)$

$Tr(Q^T Q) \geq 0$

Where Tr denotes the matrix trace. Q is a PSD since Q is symmetric since K_1 is symmetric and all other diagonal matrices are symmetric.

Thus positive semi definite.

Problem 3

A:

$z_1 = \sigma\left(\overrightarrow{w_1}^T \vec{x}\right)$
$z_2 = \sigma\left(\overrightarrow{w_2}^T \vec{x}\right)$
$y = v_1 z_1 + v_2 z_2$
$y = v_1 * \left(\sigma\left(\overrightarrow{w_1}^T \vec{x}\right)\right) + v_2 * \left(\sigma\left(\overrightarrow{w_2}^T \vec{x}\right)\right)$

B:

LMS loss function update
$W_1 = W + \eta(-\Delta j \vec{x})$
$= \begin{bmatrix}0\\0\end{bmatrix} + \ -\left(\sigma'\left(\overrightarrow{w_1}^T \vec{x} + \overrightarrow{w_2}^T \vec{x}\right) * \left(v_1 \sigma\left(\overrightarrow{w_1}^T \vec{x}\right)(y - \hat{y}) + v_2 \sigma\left(\overrightarrow{w_2}^T \vec{x}\right)(y - \hat{y})\right)\right) * \vec{x}$
$= \begin{bmatrix}0\\0\end{bmatrix} + \ \vec{x}$
$W_1 = (\vec{x})$
After 1 iteration

C:

Looking at the update function with $v_1 = v_2 = 0$ We see that most of the $-\Delta j$ element goes away. Which leaves the update to be multiples of $\vec{x}$ as it is $-\Delta j \ * \vec{x}$

For example W_2 after 1 iteration would also be $(\vec{x})$ according to the above.

Thus the relationship is that $\overrightarrow{w_1} = \overrightarrow{w_2}$ after n iterations.

D:

This is a good way to initialize these parameters. This is because the intuition of update is that with positive error we increase the weights. We increase the weights each time by the respective feature in the $\vec{x}$ vector. It does require a better learning parameter. Since this update currently may be too drastic to begin converging.
A smaller η should be chose if the parameters are initialized this way.

Problem 4

A:

$L(\vec{x};\vec{\lambda},\vec{q},P) = \vec{x}^T P\vec{x} + \vec{q}^T\vec{x} + \vec{\lambda}^T(A\vec{x} - \vec{a})$

$\nabla_{\vec{x}} = 2P\vec{x} + \vec{q} + A^T\vec{\lambda} = 0$

$2P\vec{x} = -\left(\vec{q} + A^T\vec{\lambda}\right)$

$$\vec{x} = -\frac{\left(P^{-1}\left(\vec{q} + A^T\vec{\lambda}\right)\right)}{2}$$

$$\vec{x} = -\frac{1}{2}\left(P^{-1}\vec{q} + P^{-1}A^T\vec{\lambda}\right)$$

$D(\vec{\lambda}) = \frac{1}{4}\left(P^{-1}\vec{q} + P^{-1}A^T\vec{\lambda}\right)P\left(P^{-1}\vec{q} + P^{-1}A^T\vec{\lambda}\right) - \frac{1}{2}\vec{q}^T\left(P^{-1}\vec{q} + P^{-1}A^T\vec{\lambda}\right) - \frac{1}{2}\vec{\lambda}^T A\left(P^{-1}\vec{q} + P^{-1}A^T\vec{\lambda}\right) - \vec{\lambda}\vec{a}$

$$D(\vec{\lambda}) = \frac{1}{4}\left(P^{-1}\vec{q} + P^{-1}A^T\vec{\lambda}\right)\left(\vec{q} + A^T\vec{\lambda}\right) - \frac{1}{2}\vec{q}^T\left(P^{-1}\vec{q} + P^{-1}A^T\vec{\lambda}\right) - \frac{1}{2}\vec{\lambda}^T A\left(P^{-1}\vec{q} + P^{-1}A^T\vec{\lambda}\right) - \vec{\lambda}\vec{a}$$

$$D(\vec{\lambda}) = \frac{1}{4}\left(P^{-1}\vec{q} + P^{-1}A^T\vec{\lambda}\right)\left(\left(\vec{q} + A^T\vec{\lambda}\right) - 2\vec{q}^T - 2\vec{\lambda}^T A\right) - \vec{\lambda}\vec{a}$$

The LaGrange multiplier $\vec{\lambda}$ is the $\arg\max_{\vec{\lambda}} D(\lambda)$ of which $D(\vec{\lambda})$ is shown above. $\vec{\lambda} \in \mathrm{R}^k$

B:

A closed form is possible. The closed form is the $\min_{\vec{x}} L(\vec{x};\vec{\lambda}) = D(\vec{\lambda})$

Which we have solved in the previous problem

$$D(\vec{\lambda}) = \frac{1}{4}\left(P^{-1}\vec{q} + P^{-1}A^T\vec{\lambda}\right)\left(\left(\vec{q} + A^T\vec{\lambda}\right) - 2\vec{q}^T - 2\vec{\lambda}^T A\right) - \vec{\lambda}\vec{a}$$

C:

```
Given a data set of features $\{\overrightarrow{x_1}, \overrightarrow{x_2}, \dots\}$ and labels $\vec{y}$
Initialize weights $\vec{w}$ and constant $b$
Parameters $\vec{a}$
*Assume P exists for the algorithm
Take the Lagrangian Function $L(\vec{x};\vec{\lambda},\vec{q},P)$ and substitue in
$L(\vec{w},\vec{b};\vec{a}) = b^2\vec{w}^T P\vec{w} + \vec{a}^T\vec{w} + \vec{\alpha}$
Now make the $D(\vec{\alpha}) = \frac{1}{4}(P^{-1}\vec{\alpha})\left(\left|\left|\vec{w}\right|\right|\right)^2\left(\left|\left|\vec{\alpha}\right|\right|\right)^2 - \vec{\alpha}$
Directly compute the slackness
    if $\alpha_i > 0 \rightarrow y_i(\vec{w}^T\overrightarrow{x_i} + b) = 1$
        $(\overrightarrow{x_i}, y_i)$ is a support vector
    if $\alpha_i = 0 \rightarrow y_i(\vec{w}^T\overrightarrow{x_i} + b) > 1$
        $(\overrightarrow{x_i}, y_i)$ is not a support vector
```