Shadeeb Hossain **Homework -7** **ID: sh7492**

Number -1

① $J = z_1 e^{z_1 z_2}$, where

$z_1 = a_1 w_1 w_2$.

$z_2 = a_2 w_1 + a_3 w_2^2$

(a) Partial Derivative

$\frac{\partial J}{\partial w_j}$ where $j = 1 \& 2$

So Initially According to Product rule & chain rule.

$$\boxed{\frac{\partial J}{\partial w_1}} = \left(\frac{\partial J}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_1}\right) + \left(\frac{\partial J}{\partial z_2} \cdot \frac{\partial z_2}{\partial w_1}\right)$$

$$\frac{\partial J}{\partial z_1} = \frac{\partial}{\partial z_1}(z_1 e^{z_1 z_2}) = \boxed{z_2 e^{z_1 z_2}}$$

$$\frac{\partial J}{\partial z_2} = \frac{\partial}{\partial z_2}(z_1 e^{z_1 z_2}) = \boxed{z_1 e^{z_2 z_1}}$$

$$\frac{\partial z_1}{\partial w_1} = \frac{\partial}{\partial w_1}(a_1 w_1 w_2) = \boxed{a_1 w_2}$$

$$\frac{\partial z_2}{\partial w_1} = \frac{\partial}{\partial w_1}(a_2 w_1 + a_3 w_2^2) = \boxed{a_2}$$

$$\boxed{\frac{\partial J}{\partial w_1}} = z_2 e^{z_1 z_2} \cdot a_1 w_2 + a_2 (z_1 e^{z_2 z_1})$$ Ans//

Similarly,

$$\boxed{\frac{\partial J}{\partial w_2}} = \left(\frac{\partial J}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_2}\right) + \left(\frac{\partial J}{\partial z_2} \times \frac{\partial z_2}{\partial w_2}\right)$$

$$\frac{\partial z_1}{\partial w_2} = \frac{\partial}{\partial w_2}(a_1 w_1 w_2)$$

$$= \boxed{a_1 w_1}$$

$$\frac{\partial z_2}{\partial w_2} = \frac{\partial}{\partial w_2}(a_2 w_1 + a_3 w_2^2)$$

$$= \boxed{2 a_3 w_2}$$

$$\boxed{\frac{\partial J}{\partial w_2}} = \left(z_2 e^{z_1 z_2} \cdot a_1 w_1\right) + \left(z_1 e^{z_1 z_2} \cdot 2 a_3 w_2\right) \boxed{\text{Ans}}$$

```
# Homework -1(b)

def Jeval(w):

    # Unpack vector
     w1, w2 = w
    a1, a2, a3 = a


    # Compute the loss function
    yerr = y-z1*np.exp(-z1*z2)
    J = 0.5*np.sum(yerr**2)

    # Compute the gradient
    dJ_dw1 = np.exp(z1 * z2) * (z2 * a1 * w2 + z1 * a2)
    dJ_dw2 = np.exp(z1 * z2) * (z2 * a1 * w1 + z1 * a3 * w2)

    gradient = np.array([dJ_dw1, dJ_dw2])

# Generate some random data
ny = 100
y = np.random.randn(ny)
x = np.random.rand(ny)

# Some arbitrary parameters
# to compute the gradient at
z1 = 1
z2 = 2
a1=1
a2=4
a3=7


J, Jgrad = Jeval(w)
print('Jgrad = ' + str(Jgrad))
```

Number 2

2(a) $$J(w,b) = \sum_{i=1}^{N} (\log(y_i) - \log(\hat{y}_i))^2$$

$$\hat{y}_i = \sum_{j=1}^{p} x_{ij} w_j + b$$

$$\frac{\partial J}{\partial w_j} = \frac{\partial}{\partial w_j} \sum_{i=1}^{N} (\log(y_i) - \log(\hat{y}_i))^2$$

$$= 2 \frac{\partial}{\partial w_j} \sum_{i=1}^{N} (\log(y_i) - \log(\hat{y}_i))$$

$$\cancel{= 2 \sum_{i=1}^{N} [\log(y_i) - \log(\hat{y}_i)]}$$

According to chain rule.

$$\frac{\partial}{\partial w_j} (\log \hat{y}_i) = \frac{\partial}{\partial w_j} \sum_{j=1}^{p} x_{ij} w_j + b$$

$$= x_{ij} + 0$$

$$\boxed{\frac{\partial J}{\partial w_j} = 2 \sum_{i=1}^{N} \log(y_i) - \log(\hat{y}_i) \cdot x_{ij}}$$

2(b) $$\frac{\partial J}{\partial b} = \frac{\partial}{\partial b} \sum_{i=1}^{N} (\log(y_i) - \log(\hat{y}_i))^2$$

$$= 2 \sum_{i=1}^{N} (\log(y_i) - \log(\hat{y}_i)) \cdot \frac{\partial}{\partial b} \log(y_i)$$

$$= 2 \sum_{i=1}^{N} (\log y_i - \log \hat{y}_i) \cdot \frac{\partial}{\partial b} (\cancelto{0}{x_{ij} w_j} + b)$$

$$\boxed{\frac{\partial J}{\partial b} = 2 \sum_{i=1}^{N} \log(y_i) - \log(\hat{y}_i) \cdot 1}$$

```
# Number -2(b)

import numpy as np

def Jeval(w, b, X, y):

    y_hat = np.dot(X, w) + b


    log_loss = np.log(y) - np.log(y_hat)


    J = np.sum(log_loss**2)


    Jgradw = -2 * np.sum(log_loss[:, np.newaxis] * X, axis=0)  # Calculating dj/dw


    Jgradb = -2 * np.sum(log_loss)   # calculating dj/db

    return J, Jgradw, Jgradb
```

Shadeeb Hossain **Homework -7** **ID: sh7492**

Number -3

3 (a) $$J(w) = \sum_{i=1}^{N} \left[y_i - \frac{1}{w_0 + \sum_{j=1}^{d} w_j x_{ij}} \right]^2$$

$$z_i = w_0 + \sum_{j=1}^{d} w_j x_{ij}$$

$$\frac{\partial J}{\partial w_j} = \frac{\partial}{\partial w_j} \sum_{i=1}^{N} \left[y_i - \frac{1}{w_0 + \sum_{j=1}^{d} w_j x_{ij}} \right]^2$$

According to chain rule

$$= 2 \sum_{i=1}^{N} \left(y_i - \frac{1}{z_i} \right) \cdot \frac{\partial}{\partial w_j} \left(y_i - \frac{1}{w_0 + \sum_{j=1}^{d} w_j x_{ij}} \right)$$

$$= 2 \sum_{i=1}^{N} \left(y_i - \frac{1}{z_i} \right) \cdot \left(0 - \frac{1}{0 + x_{ij}} \right)$$

$$\boxed{\frac{\partial J}{\partial w_j} = -2 \sum_{i=1}^{d} \left(y_i - \frac{1}{z_i} \right) \cdot \left(\frac{1}{x_{ij}} \right) \quad \text{Ans} \qquad \text{for } j = 1 \text{ -- } d}$$

For $$\frac{\partial J}{\partial w_0} = \frac{\partial}{\partial w_0} \left[\sum_{i=1}^{d} y_i - \frac{1}{w_0 + \sum_{j=1}^{d} w_j x_{ij}} \right]^2$$

$$\boxed{\frac{\partial J}{\partial w_0} = 2 \sum_{i=1}^{d} \left(y_i - \frac{1}{\sum_{j=1}^{d} w_j x_{ij}} \right) \quad \text{Ans}}$$

Shadeeb Hossain **Homework -7** **ID: sh7492**

```
# Number 3(b)


import numpy as np

def Jeval(w, X, y):
    n, d = X.shape
    w0 = w[0]
    wj = w[1:]


    zi = w0 + np.dot(X, wj)

    residuals = y - zi

    J = np.sum(residuals**2)

    Jgrad0 = -2 * np.sum(residuals)     # This is for dJ/dwo

    Jgradj = -2 * np.dot(residuals, X)    # This is for dJ/dwj


    Jgrad = np.hstack((Jgrad0, Jgradj))

    return J, Jgrad
```

Shadeeb Hossain **Homework -7** **ID: sh7492**

Number -4

Ans 4 (a)

$$J(a,b) = \sum_{i=1}^{N} \log(1+e^{z_i}) - y_i z_i$$

$$z_i = \sum_{j=1}^{d} a_j e^{-(x_i-b_i)^2/2}$$

$$\frac{\partial J}{\partial a_j} = \frac{\partial}{\partial a_j} \sum_{i=1}^{N} \log(1+e^{z_i}) . - y_1 z_i$$

$$= \sum_{i=1}^{N} \log(1+e^{z_i}) - y_i \cancel{z} \cdot \frac{\partial}{\partial a_j}\left(\sum_{j=1}^{d} a_j e^{-(x_i-b_i)^2/2}\right)$$

$$\boxed{\frac{\partial J}{\partial a_j} = \left[\sum_{i=1}^{N} \log(1+e^{z_i}) - y_i\right] . \ e^{-(x_i-b_i)^2/2} \quad \text{Ans}}$$

$$\frac{\partial J}{\partial b_j} = \frac{\partial}{\partial b_j} \sum_{i=1}^{N} \log(1+e^{z_i}) - y_1 z_i$$

$$= \sum_{i=1}^{N}\left[\log \cdot (1+e^{z_i}) - y_i\right] . \frac{\partial}{\partial b_j}\left(a_j e^{-(x_i-b_i)^2/2}\right)$$

$$\boxed{\frac{\partial J}{\partial b_j} = \left[\sum_{i=1}^{N} \log(1+e^{z_i}) - y_i\right]\left[a_j . \cancel{2} \frac{1}{\cancel{2}} e^{-(x_i-b_i)^2/2} \cdot (-x_i-b_j)\right.}$$

Shadeeb Hossain **Homework -7** **ID: sh7492**

```python
# Answer -4b


import numpy as np

def Jeval(a, b, X, y):
    N, d = X.shape
    z = np.zeros(N)


    for i in range(N):
        for j in range(d):
            z[i] += a[j] * np.exp(-(X[i] - b[j])**2 / 2)


    log_odds = np.log(1 + np.exp(z))


    J = np.sum(log_odds - y * z)


    Jgrada = np.zeros(d)         # Calculating dJ/da
    for j in range(d):
        Jgrada[j] = np.sum((1 - 1 / (1 + np.exp(z))) * np.exp(-(X - b[j])**2 / 2) * a[j])


    Jgradb = np.zeros(d)            # Calculating dJ /db
    for j in range(d):
        Jgradb[j] = np.sum((1 - 1 / (1 + np.exp(z))) * np.exp(-(X - b[j])**2 / 2) * a[j] * (X - b[j]))

    return J, Jgrada, Jgradb
```

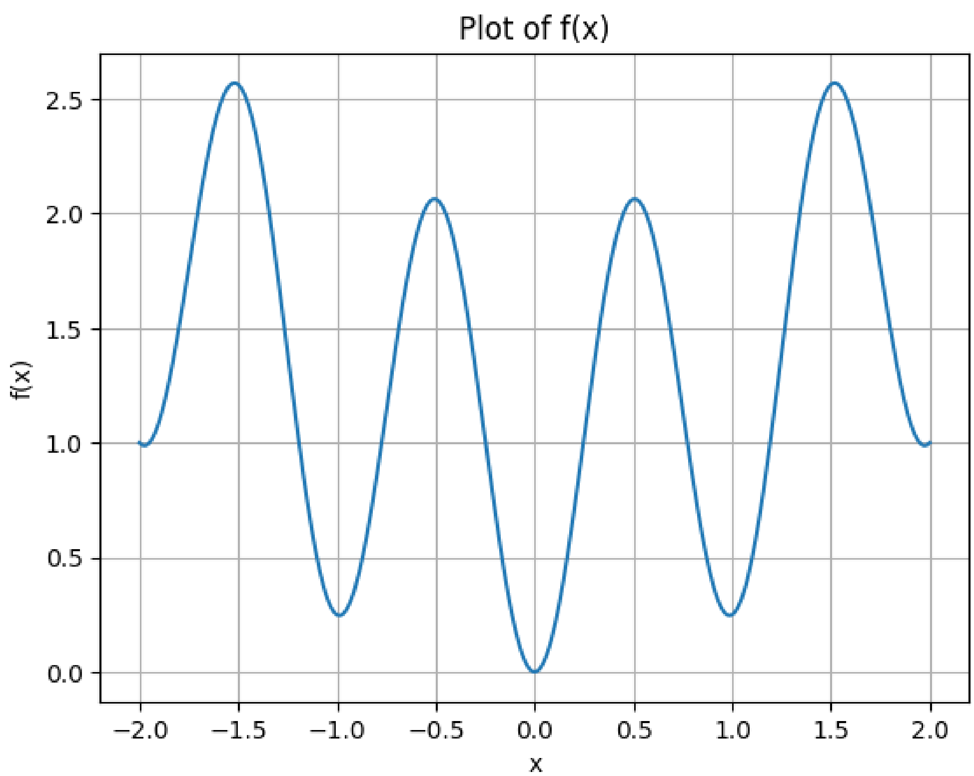
Shadeeb Hossain **Homework -7** **ID: sh7492**

Number -5

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-2, 2, 400)
f_x = 1/4 * x**2 + 1 - np.cos(2 * np.pi * x)

plt.plot(x, f_x)
plt.xlabel('x')
plt.ylabel('f(x)')
plt.title('Plot of f(x)')
plt.grid(True)
plt.show()
```



Shadeeb Hossain **Homework -7** **ID: sh7492**

5(b) $f(x) = \frac{1}{4}x^2 + 1 - \cos(2\pi x)$

$-\nabla_w f(w)$ = points in the direction of the steepest decrease.

gradient descent Algorithm

$$w^{k+1} = w^k - \underbrace{\alpha_k}_{\text{step size.}} \nabla f(w^k)$$

$$\boxed{x^{k+1} = x^k - \underbrace{\alpha_k}_{\text{step size}} \nabla f(x^k)}$$

$$x^{k+1} = x^k - \alpha_k \frac{\partial}{\partial x}\left[\frac{1}{4}x^2 + 1 - \cos(2\pi x)\right]$$

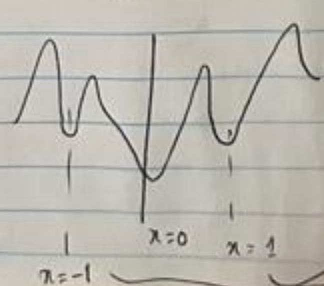
$$x^{k+1} = x^k - \alpha_k \left[\frac{1}{4} \times 2\, x + 0 - 2\pi \sin(2\pi x)\right]$$

where α_k is the step size

$$\boxed{x^{k+1} = x^k - \underbrace{\alpha_k}_{\text{step size} = 0.001} \left[\frac{1}{2}x - 2\pi \sin(2\pi x)\right]}$$ Ans

5c The global minimum is at $x=0$.

5d



From the graph it is evident one of the initial conditions could be either $\boxed{x>1 \text{ or } x<-1}$ and depending of α_k the global minimum of $x=0$ can be achieved

Shadeeb Hossain **Homework -7** **ID: sh7492**

Number -6

6 (a) $J(w) = \frac{1}{2} b_1 w_1^2 + \frac{1}{2} b_2 w_2^2$

$w = (\omega_1, \omega_2)$ $\quad b_2 > b_1 > 0$

$$\nabla J(w) = \frac{\partial}{\partial w_1}\left(\frac{1}{2} b_1 w_1^2 + \frac{1}{2} b_2 w_2^2\right)$$

$$\frac{\partial J}{\partial w_1} = \frac{1}{\cancel{2}} b_1 \cdot \cancel{2} w_1 + 0$$

$$\boxed{\frac{\partial J}{\partial w_1} = b_1 w_1}$$

$$\frac{\partial J}{\partial w_2} = \frac{\partial}{\partial w_2}\left(\frac{1}{2} b_1 w_1^2 + \frac{1}{2} b_2 w_2^2\right)$$

$$= 0 + \frac{1}{\cancel{2}} \times \cancel{2}\, b_2 w_2$$

$$\boxed{\frac{\partial J}{\partial w_2} = b_2 w_2}$$

(b) $w^* = \arg\min_w J(w)$

$$\frac{\partial J}{\partial w_1} = b_1 w_1$$

$$0 = b_1 w_1$$

$$\boxed{w_1^* = 0}$$

Similarly

$$\frac{\partial J}{\partial w_2} = b_2 w_2$$

$$0 = b_2 w_2$$

$$\boxed{\therefore w_2^* = 0}$$

6(c) From gradient descent

$$w_1^{k+1} = \rho_1 \; w_1^{k}$$

$$w_2^{k+1} = \rho_2 \; w_2^{k}$$

$$\rho_1 = \rho_0 - \underbrace{\alpha}_{\text{step size}} \frac{\partial J}{\partial w_1}$$

$$\boxed{\rho_1 = \rho_0 - \alpha \; b_1 w_1}$$ → initial value.

Similarly

$$\rho_2 = \rho_0 - \alpha \frac{\partial J}{\partial w_2}$$

$$\boxed{\rho_2 = \rho_0 - \alpha \cdot b_2 w_2}$$ → initial value.

6(d) Learning rate can be any value. most commonly used are 0.001 or even 0.01 but smaller the learning rate α the more number of steps required ~~[illegible]~~

6(e)

Shadeeb Hossain **Homework -7** **ID: sh7492**

<u>Number -7</u>

7(b)

$$J(P) = \sum_{i=1}^{n} \left[\frac{z_i}{y_i} - \ln(z_i) \right] \quad , z_i = x_i^T P x_i$$

∇P_{z_i}

$$\frac{\partial J}{\partial P_i} = \frac{\partial}{\partial P_i} \left(\sum_{i=1}^{n} \frac{z_i}{y_i} - \ln(z_i) \right]$$

$$= \sum_{i=1}^{n} \left(\frac{z_i}{y_i} - \ln z_i \right) \cdot \frac{\partial}{\partial P_i} \left(x_i^T P_{x_i} \right)$$

$$\boxed{\frac{\partial J}{\partial P_i} = \sum_{i=1}^{n} \left(\frac{z_i}{y_i} - \ln z_i \right) \cdot x_i^T}$$ Ans

Shadeeb Hossain **Homework -7** **ID: sh7492**

Number -8

8(a) $J_1(w_1) = \min J(w_1, w_2)$ $\hat{w}_2(w_1) = \arg\min J(w_1, w_2)$

8(b) $J(a,b) = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} b_j \, e^{-a_j x_i}\right)^2$

$\hat{b} = \arg\min J(a,b)$

$$\frac{\partial J}{\partial b} = \frac{\partial}{\partial b} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} b_j - e^{-a_j x_i}\right)^2$$

$$= 2 \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} b_j - e^{-a_j x_i}\right) . \left(-e^{-a_j x_i}\right)$$

8(c) $$\frac{\partial J}{\partial a} = \frac{\partial}{\partial a} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} b_j \, e^{-a_j x_i}\right)^2$$

$$= 2 \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} b_j \, e^{-a_j x_i}\right) . \; 2 b_j \, e^{-a_j x_i}$$

$$\nabla_a J(a,b) = \frac{\partial J}{\partial a} \cdot \frac{\partial J}{\partial b} = 2 \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} b_j - e^{-a_j x_i}\right) . \left(-e^{-a_j x_i}\right)$$

$$\cdot \, 2 \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} b_j \, e^{-a_j x_i}\right) . \; 2 b_j \, e^{-a_j x_i}$$

Shadeeb Hossain **Homework -7** **ID: sh7492**