QUESTION 3

ABHAY SHANKAR K: CS21BTECH11001 & KARTHEEK TAMMANA: CS21BTECH11028

(I) With the cross-entropy error given by

$$E(\mathbf{w}) = -\sum_{n=1}^{N} (t_n \ln y_n + (1 - t_n) \ln(1 - y_n)) \tag{1}$$

we have

- Gradient of the error:
$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\, \phi_n = \mathbf{\Phi^T}(\mathbf{y} - \mathbf{t})$$
- The Hessian:
$$\nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} y_n(1 - y_n)\phi_n\phi_n^T = \mathbf{\Phi^T R \Phi}$$
where R is the diagonal matrix given by $R_{nn} = y_n(1 - y_n)$
- Update function:
$$\mathbf{w}^{(new)} = \mathbf{w}^{(old)} - \mathbf{H}^{-1}\nabla E(\mathbf{w}) =$$

$$\begin{aligned} \mathbf{w}^{(new)} &= \mathbf{w}^{(old)} - \mathbf{H}^{-1}\nabla E(\mathbf{w}) \\ &= \mathbf{w}^{(old)} - (\mathbf{\Phi^T R \Phi})^{-1}\mathbf{\Phi^T}(\mathbf{y} - \mathbf{t}) \\ &= (\mathbf{\Phi^T R \Phi})^{-1}\left(\mathbf{\Phi^T R \Phi w}^{(old)} - \mathbf{\Phi^T}(\mathbf{y} - \mathbf{t})\right) \\ &= (\mathbf{\Phi^T R \Phi})^{-1}\mathbf{\Phi^T R z} \end{aligned} \tag{2}$$

with
$$\mathbf{z} = \mathbf{\Phi w}^{(old)} - \mathbf{R^{-1}}(\mathbf{y} - \mathbf{t})$$

where $\mathbf{\Phi}$ is the N $\times$ M design matrix, whose n'th row is given by ϕ_n^T, and $y_n = \sigma(\mathbf{w}^T\phi_n)$.

The algorithm for update, implemented in python:

```python
import numpy as np
def update(w, Phi, t):
    y =  mp.array([sigmoid(p @ w) for p in Phi])
    R = np.diag(y * (1 - y))
    z = Phi @ w - np.linalg.inv(R) @ (y - t)
    return np.linalg.inv(Phi.T @ R @ Phi) @ Phi.T @ R @ z
```

(II)

(III) The weighing matrix R is not constant, but depends on the parameter vector $\mathbf{w}$.

Thus, we must apply the normal equations iteratively, each time using the new weight vector $\mathbf{w}$ to compute a revised weighing matrix R. So, the algorithm is known as iterative reweighted least squares (IRLS).