AI-2002 Artificial Intelligence Spring - 2023

[Assignment – 03] **[Total Marks 100]**

Instructions:

1. Submit your assignment in a **soft (Report + Code File)** and **hard form (Only Report)** within the due date and time. Late submission will not be accepted in any case.
2. Due Date: **5th MAY 2023**
3. The report should necessarily contain the discussion, comments, and conclusion about the solution. Without a report, you will not get full marks.
4. Mention your names and roll numbers clearly on your document.
5. Name your zip or other folder/file that you want to submit according to the following format: **AI_A3_RollNo_FirstName**
6. Try to solve each task of the assignment on your own.
7. No excuse or resubmission is permissible.
8. There is no restriction of the language for the programming tasks.
9. **In programming tasks, you have to implement all function yourself and also to implement any built-in function of any library for specific tasks. Then compare your results with your user defined functions and built-in functions.**

Question No. 1: Linear Regression & Gradient Descent

Download training dataset, which consists of the input X file and the corresponding output Y file. The input X file has three attributes: the living area, the number of bedrooms, and the number of floors, while the output Y file represents the house prices in response to these attributes. There are $m = 50$ training examples. Perform the following tasks,

Gradient Descent Implementation: Implement the gradient descent algorithm with a learning rate = 0.05. Visualize the results after each iteration.

$$\begin{array}{l} \text{repeat until convergence } \{ \\ \quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\ \} \end{array}$$

Question No. 2: Linear Classification with Logistic Regression

We have discussed in the class that through the threshold function we may createa linear classifier. We, then soften the hard nature of the threshold function with the help of the logistic function which provide output between 0 and 1. It may be interpreted as a probability of belonging to any class. Figure 1 demonstrate the threshold function as well as the logistic function in (a) and (b) respectively.

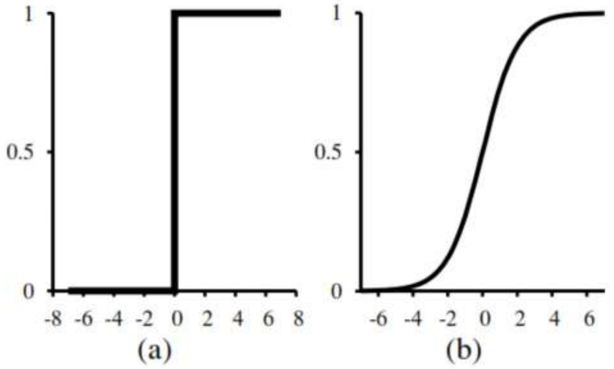

Figure 1: (a) The hard threshold function with 0/1 output. (b) The logistic function.

Your task is to implement the logistic regression in order to linearly classify the houses into two classes, “Suitable” and “Not Suitable”, using the same input data “DataX” provided in Question 1 and the classes in “ClassY” file. The “ClassY” file contains two values: **1** for “Suitable” class and **0** for “Not Suitable” class. At the end,visualize the data and the results of the logistic regression graphically. Also, discussthe results in detail.

Sample (Python) Code of Gradient Descent: Not for use, just for understanding purposes

```
import numpy as np

# Load the dataset
X = np.loadtxt('X.csv', delimiter=',')
Y = np.loadtxt('Y.csv', delimiter=',')

# Normalize the data
X_mean = np.mean(X, axis=0)
X_std = np.std(X, axis=0)
X = (X - X_mean) / X_std

# Initialize parameters
w = np.random.randn(3)
b = 0

# Define the cost function
def cost_function(X, Y, w, b):
    predictions = np.dot(X, w) + b
    cost = np.mean((predictions - Y) ** 2)
    return cost

# Train the model using Gradient Descent
learning_rate = 0.01
num_iterations = 1000

for i in range(num_iterations):
    predictions = np.dot(X, w) + b
    w = w - learning_rate * np.dot(X.T, predictions - Y) / len(Y)
    b = b - learning_rate * np.mean(predictions - Y)
    cost = cost_function(X, Y, w, b)
    if i % 100 == 0:
        print('Iteration:', i, 'Cost:', cost)

# Test the model
X_test = np.array([[1000, 2, 1], [1500, 3, 2], [2000, 4, 2]])
X_test = (X_test - X_mean) / X_std
Y_test = np.dot(X_test, w) + b
print('Predictions:', Y_test)
```