In the name of God

HW1 of Machine Learning with Python

Deadline: 30th of November (پنجشنبه ۹ آذرماه)

Email: mlfall2023@gmail.com

In this assignment, your task is to **implement the "logistic regression algorithm" in code**. I will provide the instructions in the form of questions, and you are required to write the corresponding code. Upon completion, save your file as a .ipynb file and email it before the deadline. It is crucial to note that significant penalties will be imposed if any evidence of copying from other students is detected in your implementation, resulting in a substantial reduction in your grade.

Dataset: The provided normalized dataset is about "**Sensor Network Security**":

> **Features:** Proximity and signal strength between two connected devices in a secure sensor network.
>
> **Label:** 1 (Secure connection) or 0 (Potential security breach).

Questions:

1. **Part 1:** Load the dataset provided as dataset.txt and split the data into X_train and y_train.
   Part 2: Print the first 5 elements of the X_train and y_train.
   Part 3: Check the dimensions of the X_train and y_train.
   Part 4: Visualize the data where the axes are two features and the diffirent labels are shown with different colors.
   Part 5: Use the provided **polynomial_feature_mapping** function
   to map the features into all polynomial terms of x1 and x2
   like the vector **shown on the right side**.
   Copy the function from the files and use this line of code:
   Code:
   mapped_X = polynomial_feature_mapping (X_train[:, 0], X_train[:, 1])

$$\begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \\ x_1^3 \\ \vdots \\ x_1 x_2^5 \\ x_2^6 \end{bmatrix}$$

   Part 6: Display the first element of the mapped_X

2. **Part 1:** Define a sigmoid function that takes a variable and returns its sigmoid value. The sigmoid function is defined as:

   $$g(z) = \frac{1}{1+e^{-z}}$$

   Part 2: Check your sigmoid function by evaluating this vector's sigmoid function: Vector = [-1, 0.5, 0, 1, 2]

3. **Part 1:** Define a function to compute the cost function of the logistic regression using the equation below. Make sure to define **lambda** parameter as an argument of this function.

   $$J(\mathbf{w},b) = \frac{1}{m}\sum_{i=0}^{m-1}\left[-y^{(i)}\log\left(f_{\mathbf{w},b}\left(\mathbf{x}^{(i)}\right)\right) - \left(1-y^{(i)}\right)\log\left(1-f_{\mathbf{w},b}\left(\mathbf{x}^{(i)}\right)\right)\right] + \frac{\lambda}{2m}\sum_{j=0}^{n-1} w_j^2$$

   Where:

   $$f_{\mathbf{w},b}(\mathbf{x}^{(i)}) = g(\mathbf{w}\cdot\mathbf{x}^{(\mathbf{i})} + b)$$

   And g is the sigmoid function.
   Part 2: Compute the cost of these parameters with given X_train and y_train: w1 = 0.2 , w2 = 0.2 , b = -24, lambda = 1.

4. **Part 1:** Define a function to compute the gradients using these equations:

   $$\frac{\partial J(\mathbf{w},b)}{\partial b} = \frac{1}{m}\sum_{i=0}^{m-1}(f_{\mathbf{w},b}(\mathbf{x}^{(i)}) - y^{(i)})$$

   $$\frac{\partial J(\mathbf{w},b)}{\partial w_j} = \left(\frac{1}{m}\sum_{i=0}^{m-1}(f_{\mathbf{w},b}(\mathbf{x}^{(i)}) - y^{(i)})x_j^{(i)}\right) + \frac{\lambda}{m}w_j \quad \text{for } j = 0...(n-1)$$

   Part 2: Compute and print the gradients given w1 = 0, w2 = 0, b = 0 , lambda = 1.

5. **Part 1:** Define a function that learns the parameters using gradient descent.
   You should use these arguments as the arguments of the function:
   X, y, w_in, b_in, cost_function, gradient_descent, alpha, num_iters, lambda

   Part 2: Initialize your **w** and b parameters and use your gradient descent function to find the best values for **w** and b parameters. Set the number of iterations to 10,000, the learning rate to 0.01, and the value of the lambda to 0.01 accordingly.

   Part 3: Print the cost every 1000 iteration.

6. We provide you another function called **visualize_decision_boundary.** Using this function, plot your model by this line of code:

   Code:

   visualize_decision_boundary(w, b, X_mapped, y_train)

Good Luck