Lab Course Machine Learning

Exercise Sheet 3

Prof. Dr. Dr. Lars Schmidt-Thieme
Jung Min Choi

HiWi: Harish Malik

Submission deadline : November 7, 2024

General Instructions

1. Perform a data analysis, deal with missing values and any outliers.
2. Unless explicitly noted, you are not allowed to use scikit, sklearn or any other library for solving any part.
3. Data should be normalized.
4. Train to Test split should be 80-20.
5. Convert any non-numeric values to numeric values. For example you can replace a country name with an integer value or more appropriately use one-hot encoding.

1 Linear Regression on Time Series Data (7 points)

In this part, you are required to optimize the *booth function* using gradient descent. The Figure below provides a visual representation of the *booth function* in a 3D plot.

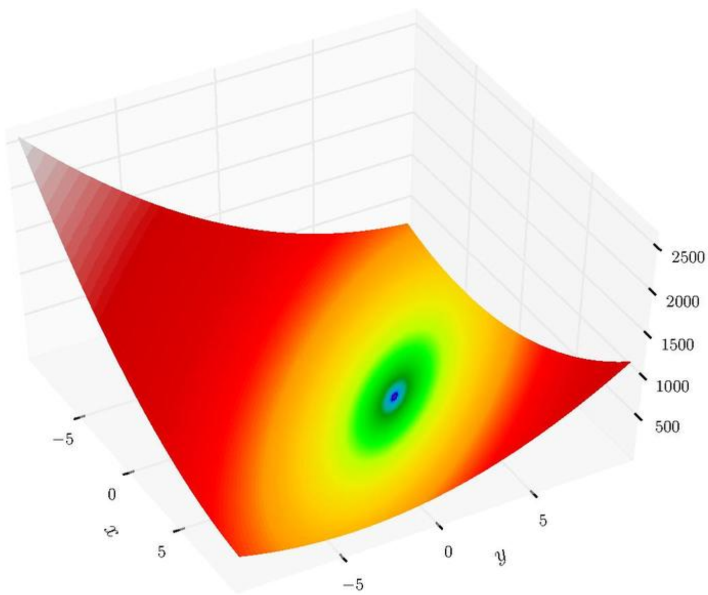

Figure 1: Booth function in a 3D plot

Mathematically, the function can be defined as follows.

$$f(x, y) = (x + 2y)^2 + (2x + y - 5)^2 \tag{1}$$

For this part of the exercise,

1. **[1 point]** Implement a 3D plot to visualize the function (Use Matplotlib's 3D utilities)
2. Derive the partial gradients.
3. **[3 point]** Optimize the function with Gradient Descent. Set the appropriate hyperparameters like initial values of (x,y) and the steplength α through trial and error.
4. **[1 point]** Visualize the trajectory on the same 3D plot. This trajectory should ideally lead to the function minimum. Try to plot the trajectory in a for loop so that the path taken is visible.
5. **[2 point]** Implement the function that controls the step length (ML1 Slides 8-10).

2 Linear and Logistic Regression. (13 points)

A [3 point] Optimization Routines and Loss Functions

1. In this part of the assignment, we learn how to write modular programs and make our code reusable. For this, declare a class named **Optimization** which has inputs `X` and `y` as the class variables. Next, implement the following optimization algorithms in this class:
   a) Newton's Method
2. You will need loss functions and their gradients for the optimization process. Next, implement a class **Loss** which also takes in `X` and `y` and computes the following losses and their gradients:
   a) Mean Square Loss
   b) Cross Entropy Loss

   Make the **Loss** class such that you can access it from the **Optimization** class.

B [5 point] Linear Regression

In this task, you are given a dataset named `regression.csv`. Implement a class **LinearRegression** that has at least two functions, **fit** and **predict**, for fitting a linear regression model and predicting the results. You need to use the **Optimization** and **Loss** classes inside this. Fit a linear regression model using the following:

1. Mean Square Loss and Newton's Method

Set the random seed to ensure that your model is reproducible.
In the end, generate the loss trajectory for both training and testing.
While optimizing the loss function for Linear Regression or Logistic Regression, one needs to initialize the model parameters. It is well known that deep neural networks do not function if the model parameters are initialized to zero. Why is it so? Does this issue also arise while optimizing the loss function for Linear or Logistic Regression? Explain.

C [5 point] Logistic Regression

You are given a file `logistic.csv`. This part of the assignment involves a classification task. Implement a class **LogisticRegression** having at least two functions, **fit** and **predict**, for fitting the model and getting the predictions. Fit a logistic regression model using the following:

1. Cross Entropy Loss and Newton's Method

Set the random seed to ensure that your model is reproducible.
Report the test accuracy, plot the confusion matrix, and also compute the precision, recall, and F-score. Read about these parameters. Suppose model A and model B both have the same accuracy, but model B has a higher F-score. Which model would be more suited? In the end, generate the loss trajectory for both training and testing.