CS412 Homework 2 Report

Uğur ÖZTUNÇ 28176

Colab Notebook Link

https://colab.research.google.com/drive/1MmzxatRn7Od1VCVmMtZrBCClVOH1Oa1P?usp=sharing

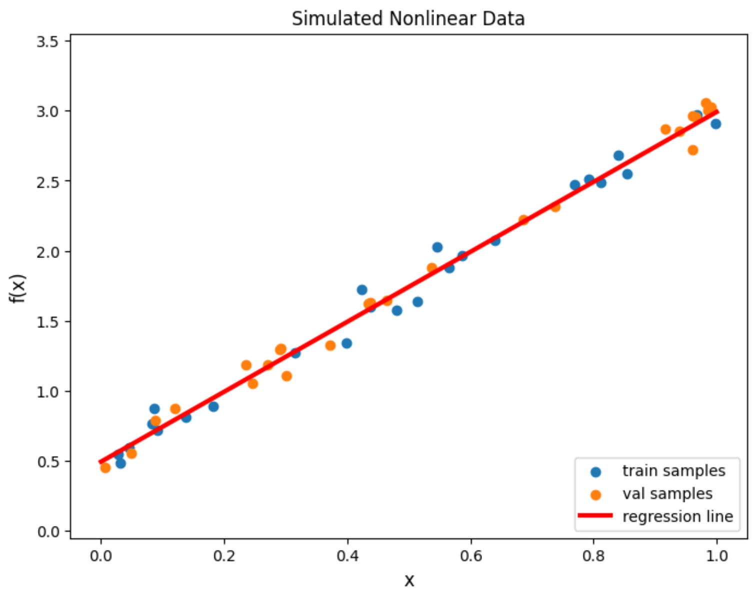
Introduction

In this assignment, we were asked to develop regression models to make predictions based on datasets that are collection of (x, y) pairs, where x values are uniformly sampled from a given range, and y values are generated by predefined linear and polynomial functions. After creating models, the performance of models on validation sets were evaluated using the mean squared error metric. The implementations of models were done both by using scikit-learn library and manually with the help of NumPy. These models were developed as follows:

1. Linear Regression Part
    a. scikit-learn library linear regression method
    b. Manually implemented OLS algorithm uses pseudoinverse method
    c. Gradient descent method
2. Polynomial Regresson Part
    a. scikit-learn library polynomial regression method
    b. Manually implemented polynomial regression method

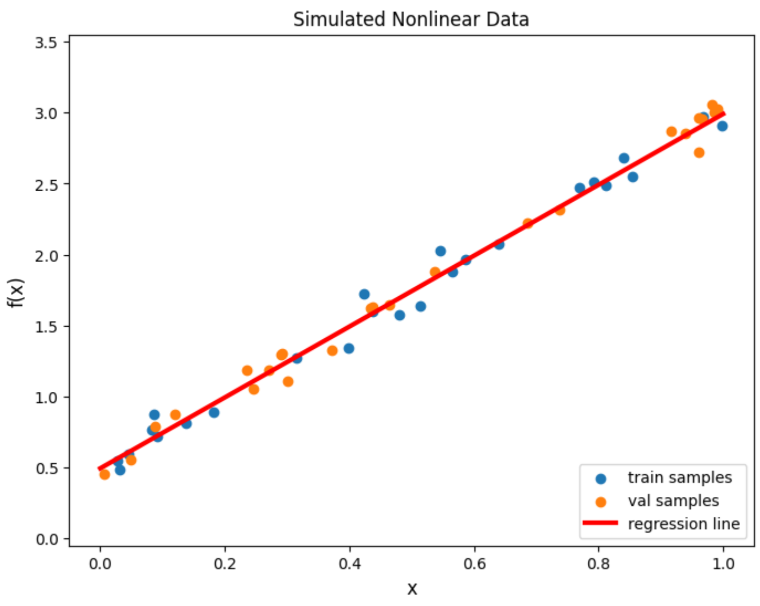
1. Linear Regression Part

a) The first model was implemented simply by using sklearn library linear regression and based on the predictions made on the validation set by the model following results found:



MSE: 0.0056786787215701

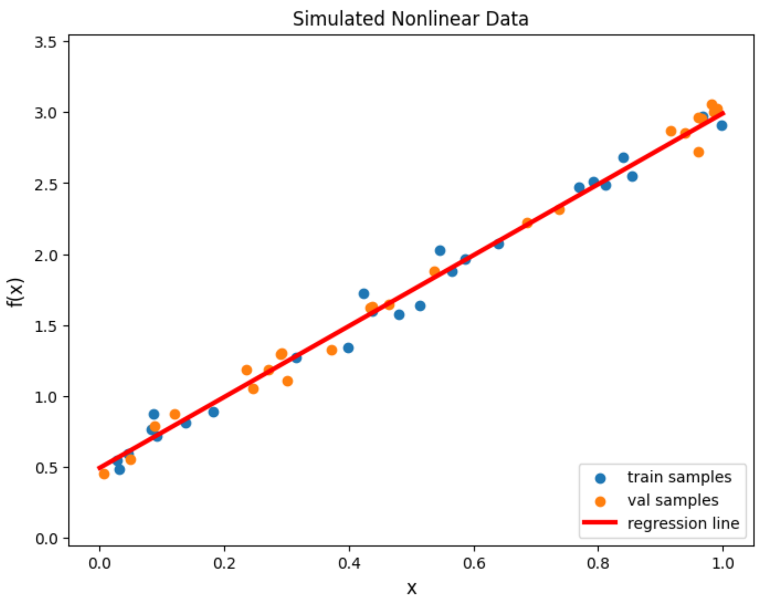
b) The second model was implemented by using pseudo-inverse method manually to find regression coefficients. At first, extended data matrices of training and validation sets were constructed. Then extended x matrix was pseudoinversed and matrix multiplication with train set y's to find coefficients. Finally, the model's prediciton was made on validation set. Here are the results:



MSE: 0.005678678721570 47
Coefficients: [2.49629849, 0.4946291]

c) In third model, the gradient descent algorithm was implemented to create a model. At first, regression coefficients were initialized randomly and gradient descent method's steps are carried out 1000 times in order to reach an optimal MSE. Here are the results:

Step	MSE
1	1.5852
100	0.2577
200	0.0595
300	0.0184
400	0.0099
500	0.0081
600	0.0077
700	0.0077
800	0.0076
900	0.0076
1000	0.0076



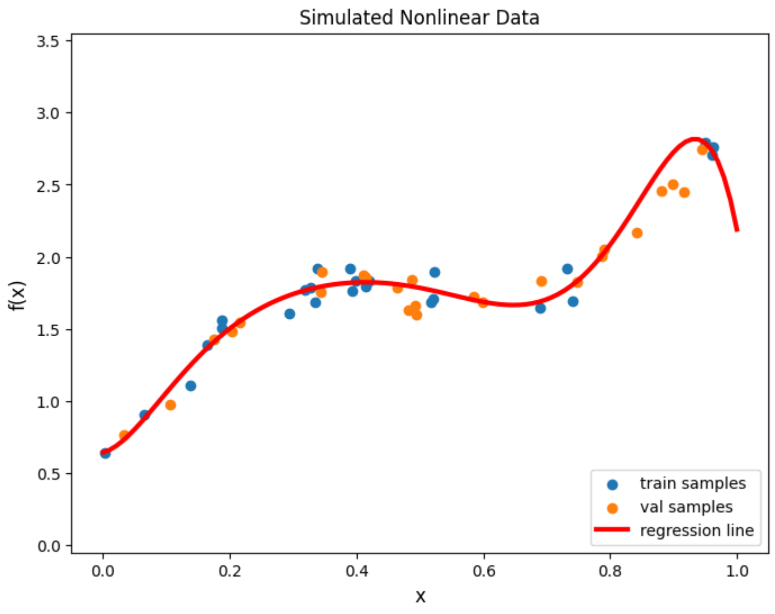
Coefficients: [2.49496588, 0.49530119]

- When we check the mean squared errors and coefficients, we see that the gradient descent algorithm is not exactly the same; however, when we look at the regression lines on figures, the two models are extremely similar.

2. Polynomial Regression Part

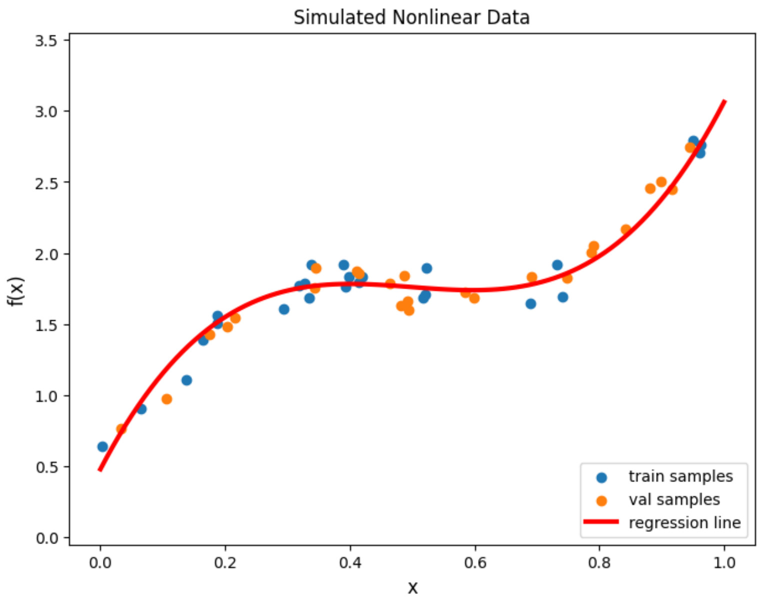
a) The first polynomial regression model was implemented by using sklearn library linear regression. Multiple models were developed with degrees 1, 3, 5, 7; and, each model was tested on validation set to see MSE. Here are the results:

Degree	MSE
1	0.047199737611932094
3	0.009143644463506495
5	0.007063512185254561
7	0.014741655064860979



Polynomial regression model with degree 7

d) The second polynomial regression model was implemented manually by creating a data matrix, which is obtained by adding the powers of the values in training set with respect to degree 3, since it resulted in the minimum MSE in previous part. Then this matrix was pseudoinversed and multiplied with train set y's to find coefficients. Finally, the model's prediciton was made on validation set. Here are the results:



Polynomial regression model with degree 3
MSE: 0.009143644463505827
Coefficients: [0.48002506, 8.37576343, -17.46498704, 11.67068266]

- The results showed that increasing the degree to some extent would improve the performance of the model, since the regression become more flexible; however, too large degree might result overfitting, as in the part a. By looking at the degree table in part a, we have determined that the degree of 3 is the most optimal one by showing the best performance with the minimum MSE, since it neither underfits as linear, nor overfits like degree 7.