Machine Learning Assignment Two (CS412)

Hagverdi Ibrahimli

30014

Link to the submission notebook:
https://colab.research.google.com/drive/1V_ZYmxLlRShqVc5OEi-zTZTY4Vhbs09E?usp=sharing

Aim of the assignment:

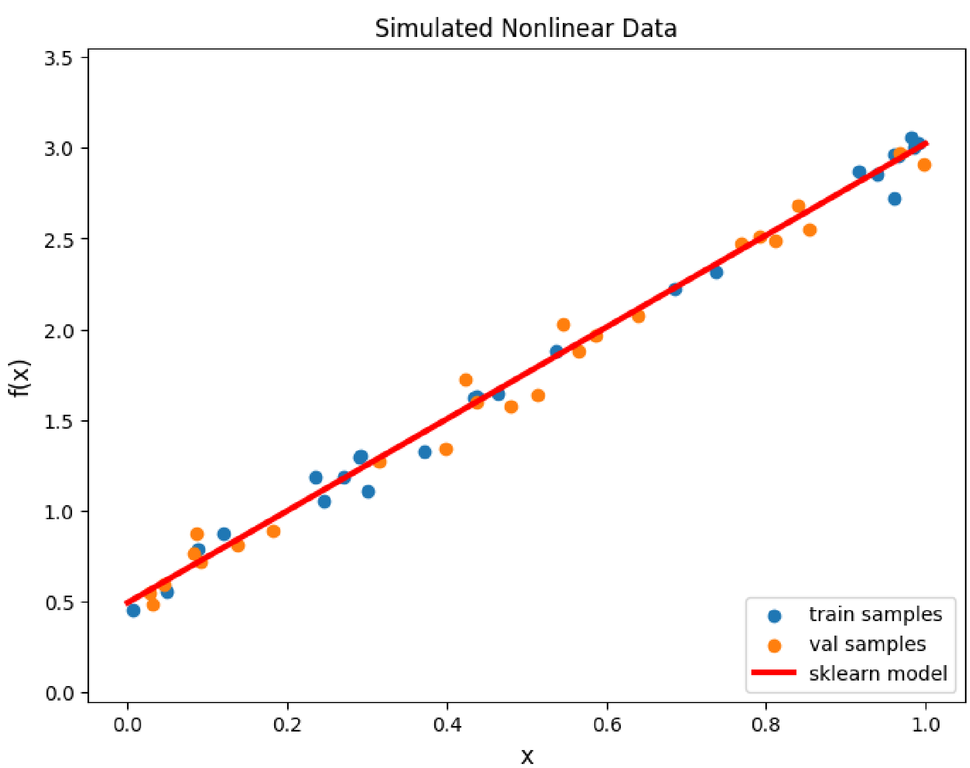
1. Introduction to linear regression, a fundamental machine learning method for modeling the relationship between variables.
2. Gain experience in using/implementing the least squares solution and gradient descent approach using NumPy.
3. Learn to apply polynomial regression to model nonlinear relationships between variables.
4. Gain experience in constructing the polynomial expansion of the data matrix and using/implementing the least squares solution.

Part 1.a Results (Sklearn Linear Regression module):

Mean squared error
0.007954

Regression coefficients	Value
W_0 (Y-Intercept)	0.494476
W_1	2.528382

Plot:

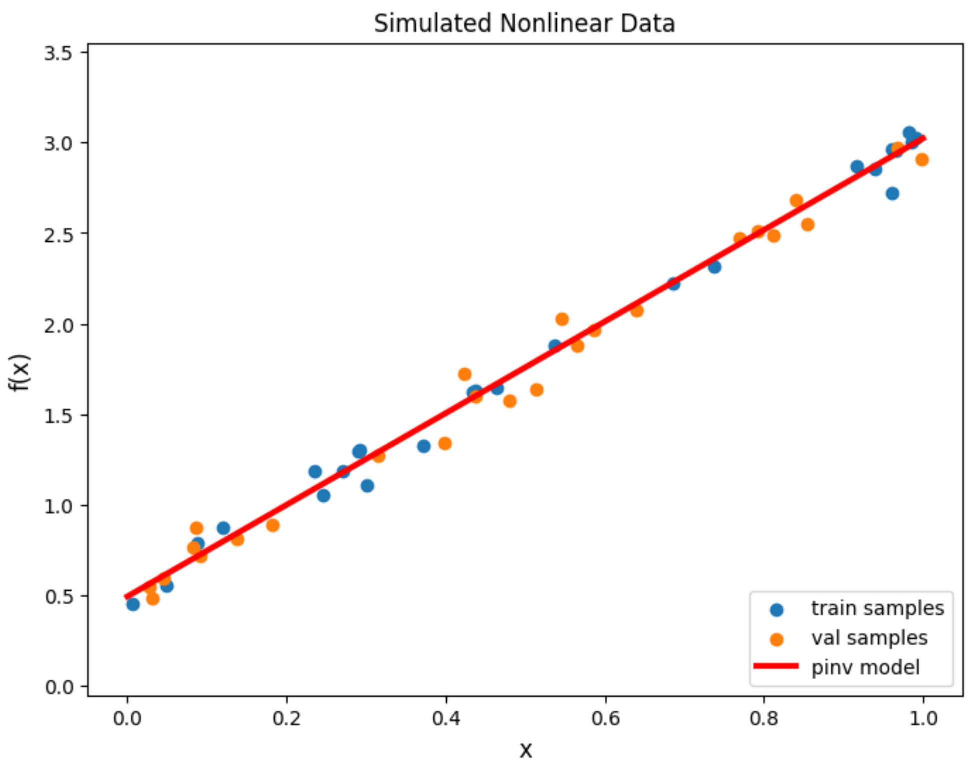


Part 1.b Results (Manual Linear Regression):

Mean squared error
0.007954

Regression coefficients	Value
W_0 (Y-Intercept)	0.494476
W_1	2.528382

Plot:

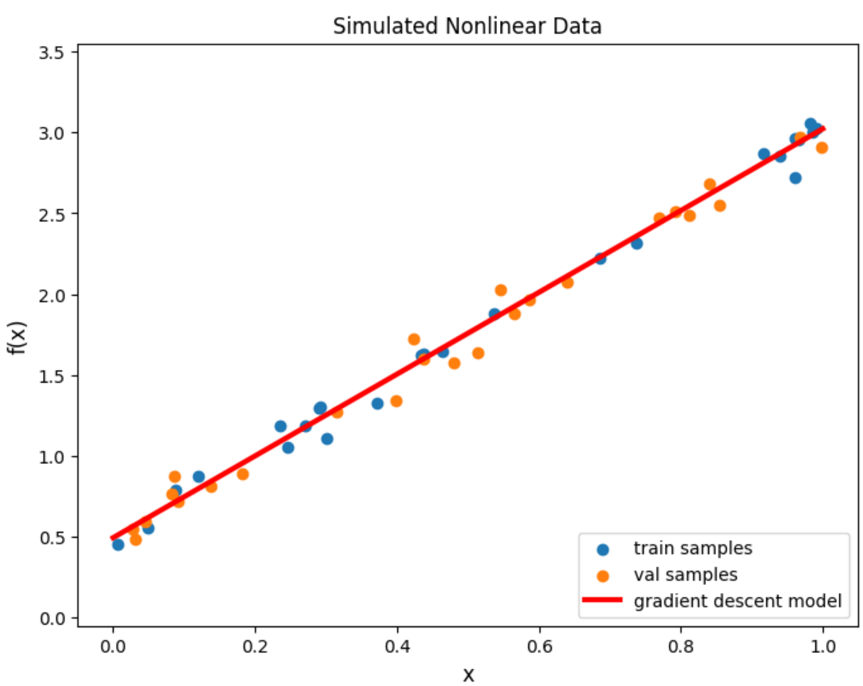


Part 1.c Results (Gradient Descent optimized model):

Step number i	Mean squared error at step number i
i = 1	4.114688
i = 100	0.06251
i = 200	0.015003
i = 300	0.006928
i = 400	0.005556
i = 500	0.005322
i = 600	0.005283
i = 700	0.005276
i = 800	0.005275
i = 900	0.005275
i = 1000	0.005275

Regression coefficients	Value
W_0 (Y-Intercept)	0.494614
W_1	2.528145

Plot:

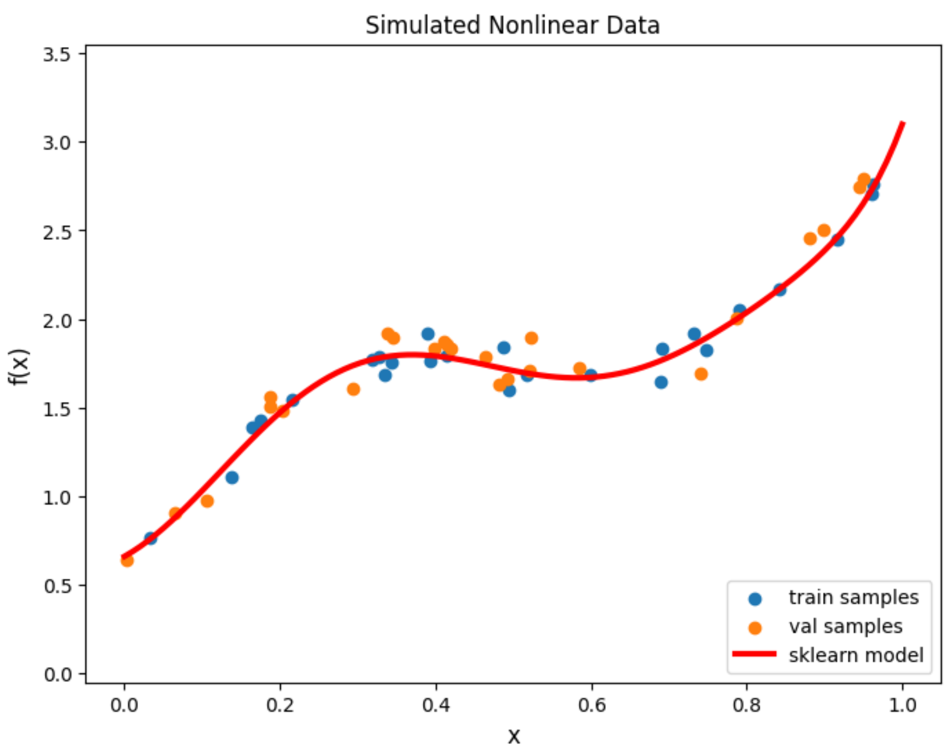


Part 2.a Results (Sklearn Polynomial Regression module):

Degree of the features	Mean squared error with features of degree n
n = 1	0.057598
n = 3	0.012732
n = 5	0.00858
n = 7	0.010406

Degree of the features	W_0 (Y-Intercept)	W_1	W_2	W_3	W_4	W_5	W_6	W_7
1	1.10983	1.352095						
3	0.432973	8.288181	-16.903953	11.21853				
5	0.62155	3.336715	16.419539	77.34796	100.531	-40.7239		
7	0.660843	2.449214	15.983496	20.36017	182.5839	525.5943	-511.3575	172.7124

Plot:

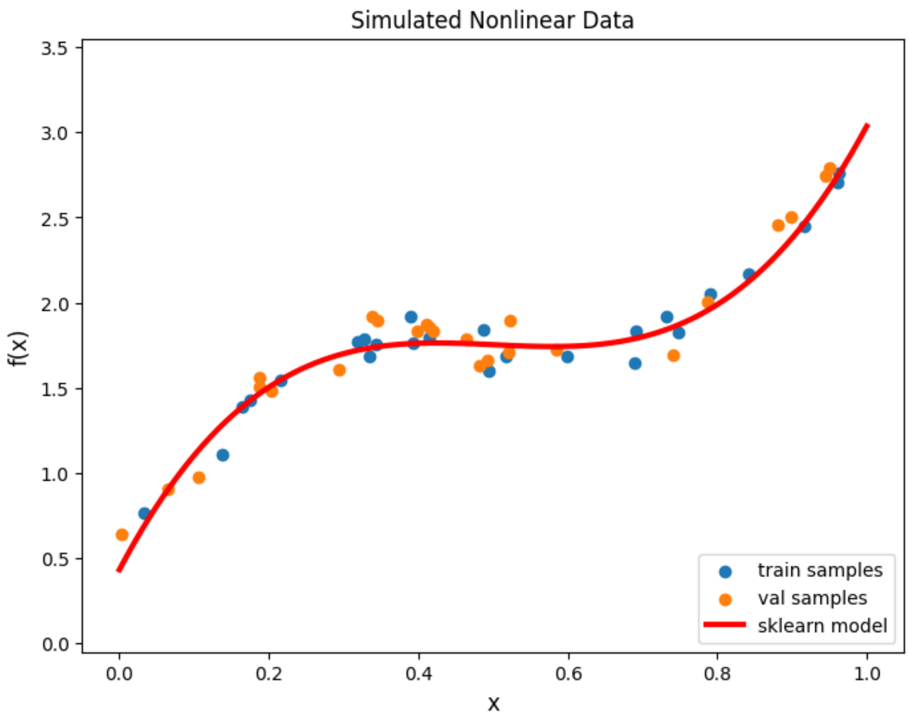


Part 2.b Results (Manual Polynomial Regression module):

Mean squared error
0.012732

Regression coefficients	Value
W_0 (Y- Intercept)	0.432973
W_1	8.288181
W_2	-16.903953
W_3	11.218532

Plot:



Comments for part 1:

Comparing the performance of the three models described in part one, gradient descent optimized model is slightly more accurate on the validation dataset (according to mean squared error metric) than the other two linear regression models. One possible reason for this difference could be that the gradient descent algorithm is more flexible than the closed-form solutions used in Part 1a and 1b. Another reason for the difference in MSE could be the presence of noise in the data. The gradient descent algorithm may have found a set of regression coefficients that better captures the noise in the training data, resulting in a lower MSE on the validation set.

Comments for part 2:

When the degree is too small, for example, degree=1, the model would not be able to capture the non-linearities in the data. In other words, the model would underfit the data. This can be seen in the fact that the mean squared error value would be relatively high.

On the other hand, when the degree is too large, for example, degree=7, the model may capture too much of the noise in the data. In other words, the model would overfit the data. Overfitting occurs when a model is too complex relative to the amount of training data, which leads to good performance on the training set but poor generalization to new data. In this case, the model would have very low mean squared error values for the training set, but higher mean squared error values for the validation set.

In our case, the model with the degree of 5 proved to be more accurate on the validation dataset than the other models with degrees 1, 3, and 7 with a mean squared error score of 0.008580.

The optimal degree value depends on the complexity of the data and the amount of training data available. A higher degree polynomial would provide more flexibility to the model to capture complex patterns in the data, but if there is not enough data to support such a complex model, it may overfit the data. A lower degree polynomial would provide a more constrained model that may not capture the complexity in the data.