CS412 - Machine Learning: Homework 2

Notebook Link: Google Colab Notebook

Title: Linear and Polynomial Regression Analysis

Name: Mete Kerem Berk
Student ID: 30933
Course: CS412 - Machine Learning
Homework Number: HW2
Submission Date: March 16, 2025

1. Generate Data for Regression

- Created a dataset of (x, y) pairs where y is generated from a linear function with added Gaussian noise.
- Saved the dataset for further use in regression tasks.

2. 50% Train 50% Validation Split

- The dataset was split into 50% training and 50% validation sets to evaluate model performance effectively.

3. Make a Scatter Plot of the Data

Scatter Plot of Dataset 1:

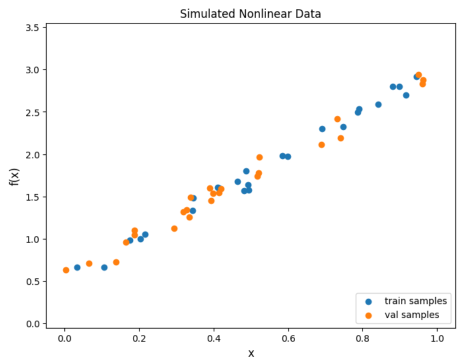

Figure 1: Scatter Plot of Dataset 1

Scatter Plot of Dataset 2:

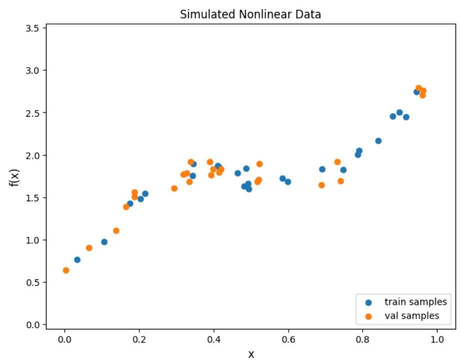

Figure 2: Scatter Plot of Dataset 2

4. Function for Plotting the MSE Loss

- Implemented a function to visualize the loss curve during training for gradient descent.

5. Part 1: Linear Regression on Dataset 1

Part 1.a - Scikit-Learn's Linear Regression

- Utilized `LinearRegression` from `sklearn.linear_model`.
- Trained the model on Dataset 1 and evaluated predictions on the validation set.

Regression Line Fit (Scikit-Learn):

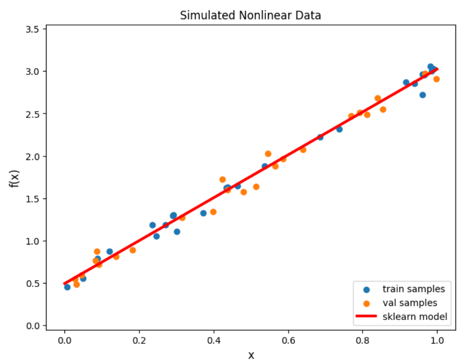

Figure 3: Regression Line Fit (Scikit-Learn)

Part 1.b - Ordinary Least Squares (OLS)

- Implemented OLS manually using matrix computations.
- Computed coefficients and used them for prediction.

Regression Line Fit (OLS):

Part 1.c - Gradient Descent

- Implemented gradient descent to iteratively optimize weights.
- Used a learning rate of 0.1 and 1000 iterations.
- Converged to optimal parameters and evaluated performance.

Regression Line Fit (Gradient Descent):

Loss Curve for Gradient Descent:

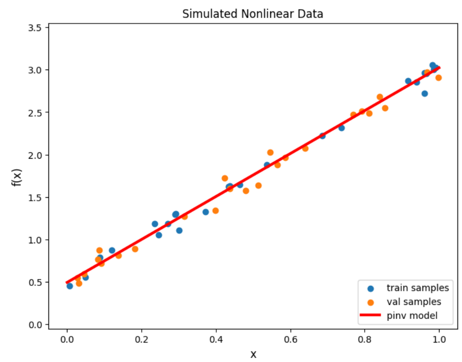

Figure 4: Regression Line Fit (OLS)

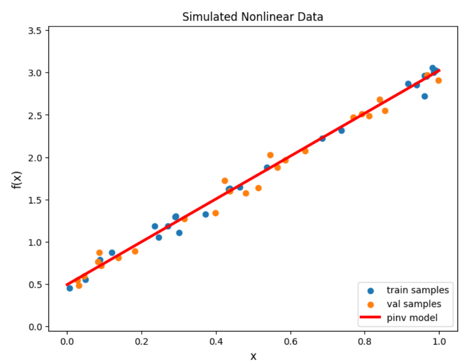

Figure 5: Regression Line Fit (Gradient Descent)

6. Part 2: Polynomial Regression on Dataset 2

Part 2 - Data Generation

- Loaded Dataset 2 from `.npy` files.
- Splitted the dataset into training and validation sets.

Part 2.a - Polynomial Regression using Scikit-Learn

- Applied `PolynomialFeatures` from `sklearn.preprocessing` to generate polynomial features of degrees 1, 3, 5, and 7.
- Fitted linear regression models to the transformed data and computed validation MSE.
- Found the best degree for the given data as 5.

Polynomial Regression Fit (Degree 5):

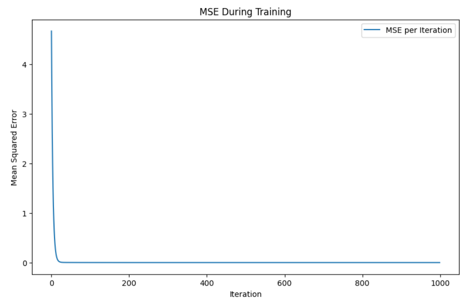

Figure 6: Loss Curve for Gradient Descent

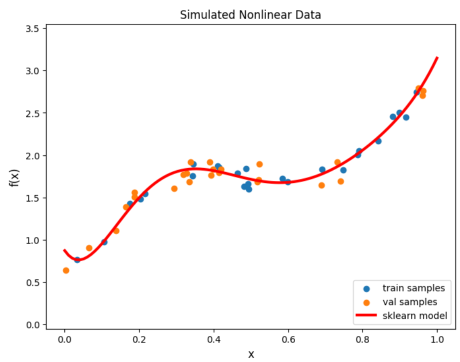

Figure 7: Polynomial Regression Fit (Degree 5)

Part 2.b - Manual Polynomial Regression

- Implemented polynomial regression manually for degree 3.
- Constructed the polynomial feature matrix and applied the OLS method.

Manual Polynomial Regression Fit (Degree 3):

7. Results & Discussion

Part 1: Comparison of Gradient Descent with Other Methods

- The gradient descent solution is very close to the solutions obtained using Scikit-Learn's Linear Regression and the manually implemented Ordinary Least Squares (OLS) method.
- Any small discrepancies can be attributed to:
  - The number of iterations chosen for gradient descent.
  - The learning rate, which affects how quickly the model converges.
  - Possible stopping conditions that might have been met before full convergence.

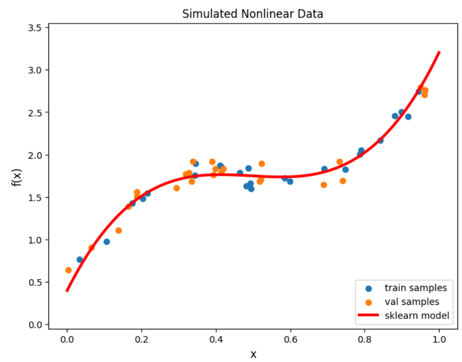

Figure 8: Manual Polynomial Regression Fit (Degree 3)

- If the gradient descent solution differs significantly, it may indicate insufficient iterations or a learning rate that is too high or too low.

Part 2: Effect of the Degree Parameter in Polynomial Regression

- When the polynomial degree is too small (e.g., degree 1), the model underfits the data and fails to capture the nonlinear relationships.
- When the polynomial degree is too large (e.g., degree 7), the model overfits, capturing noise as if it were part of the underlying pattern.
- The optimal polynomial degree balances bias and variance. Based on the MSE values, degree 5 appears to provide the best trade-off, achieving a low validation error while avoiding excessive overfitting.

Performance Comparison (MSE Analysis)

Method	MSE
Scikit-Learn Linear Regression	0.00795462682779033
Manual OLS	0.00795462682779037
Gradient Descent	0.00527514849082713
Polynomial (Degree 1)	0.06362852709262727
Polynomial (Degree 3)	0.01205922374286829
Polynomial (Degree 5)	0.00747546307928118
Polynomial (Degree 7)	0.01154922783882886

---

8. Conclusion

This homework provided hands-on experience in implementing and analyzing regression methods. Key takeaways:

- Linear regression methods performed well on dataset 1.
- Polynomial regression effectively captured nonlinearity in dataset 2, but higher degrees led to overfitting.
- Gradient descent, while powerful, required careful tuning.

Overall, this assignment reinforced the importance of selecting appropriate models based on data characteristics and balancing bias-variance tradeoffs.