ECGR 5105 Homework 1

Owen Bailey-Waltz (801488178)
GitHub Link: https://github.com/obaileyw-uncc/ecgr5105/hw01_graddesc

Problem 1: Single-variable linear regression gradient descent algorithm

1. Report the linear model you found for each explanatory variable.

Higher-precision values of each model can be found in the assignment's associated Jupyter notebook.

$$h(x_1) = -2.0x_1 + 5.9$$

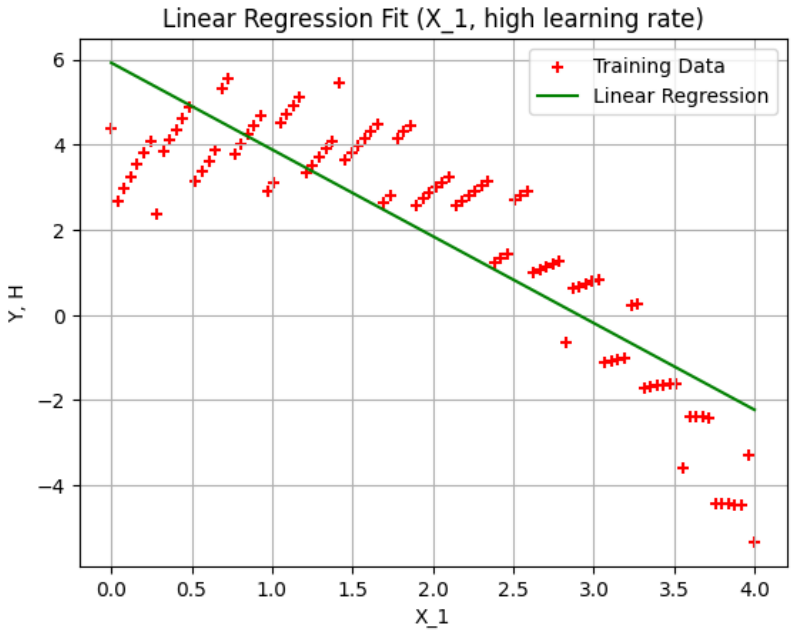
$$h(x_2) = 0.56x_2 + 0.73$$

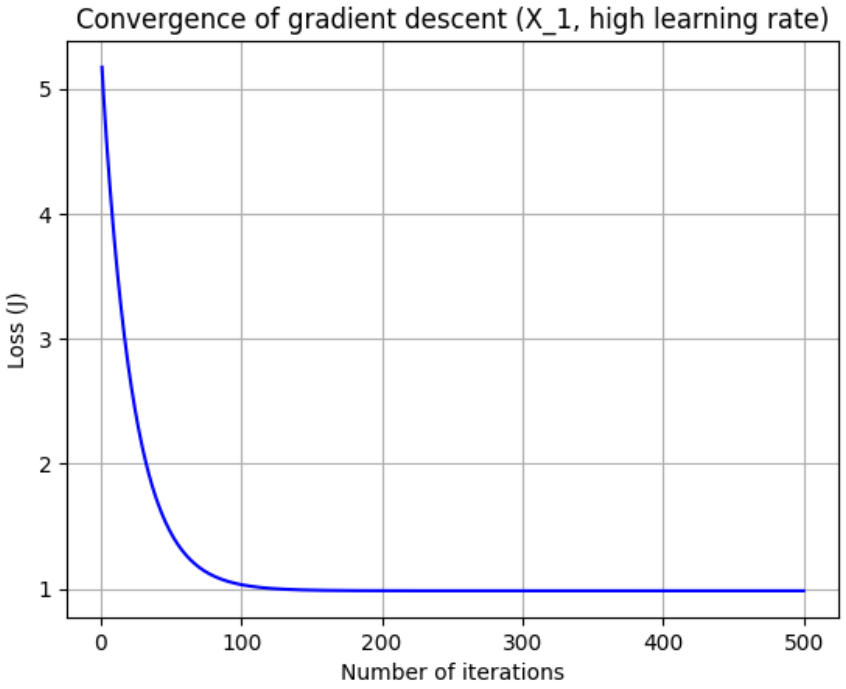
$$h(x_3) = -0.52x_3 + 2.9$$

2. Plot the final regression model and loss over the iteration for each explanatory variable.

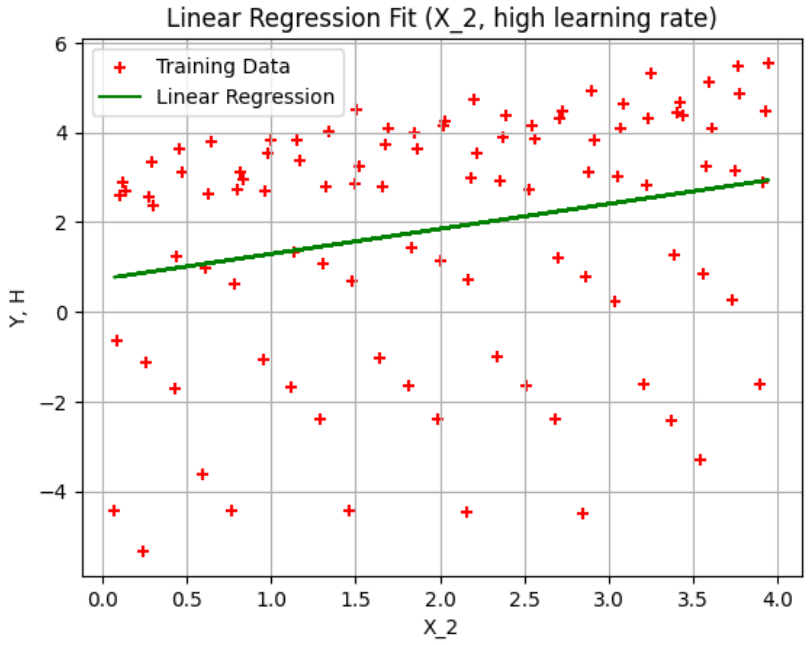
All plots are for $\alpha = 0.1$.

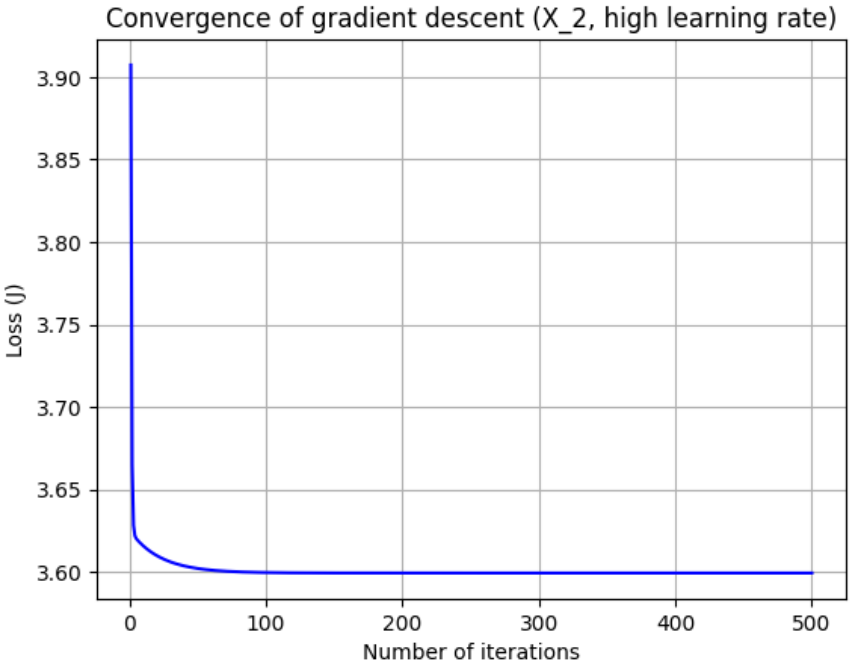
x_1



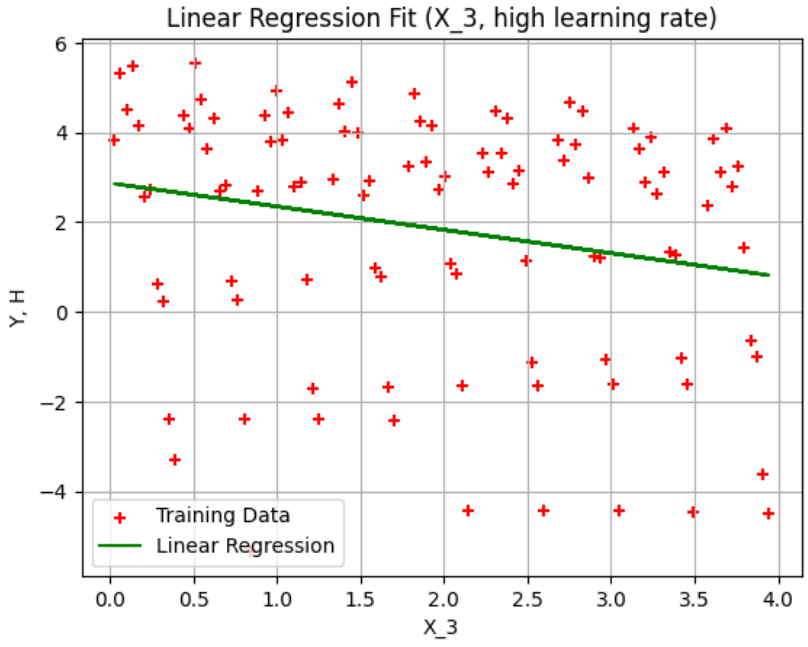

Convergence of gradient descent (X_1, high learning rate)
Loss (J)
Number of iterations
5
4
3
2
1
0
100
200
300
400
500

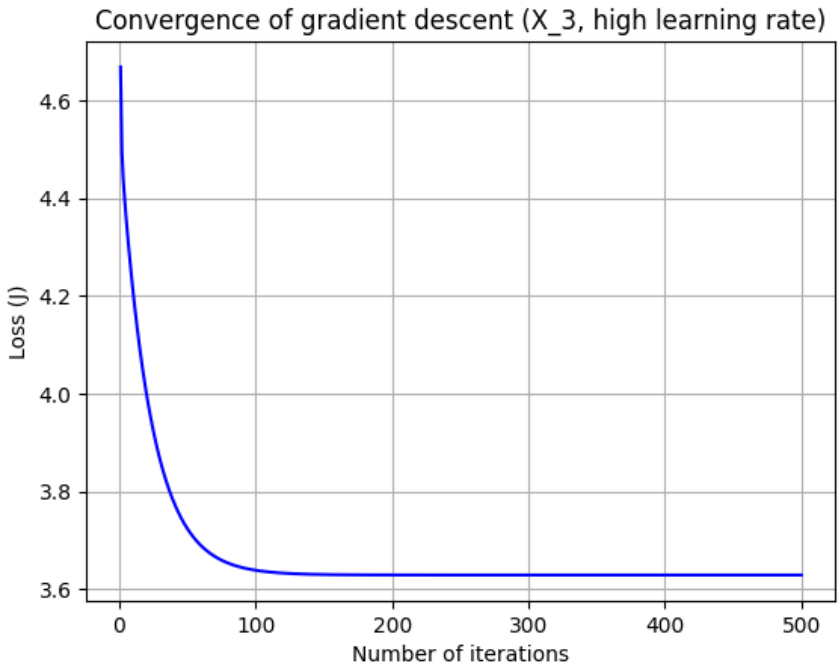
X_2


Linear Regression Fit (X_2, high learning rate)
Training Data
Linear Regression
Y, H
X_2
6
4
2
0
−2
−4
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Convergence of gradient descent (X_2, high learning rate)
Loss (J)
Number of iterations
3.90
3.85
3.80
3.75
3.70
3.65
3.60
0
100
200
300
400
500

X_3


Linear Regression Fit (X_3, high learning rate)
Y, H
X_3
Training Data
Linear Regression
6
4
2
0
−2
−4
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0



3. Which explanatory variable has the lower loss (cost) for explaining the output (Y)?

x_1 has the lowest loss for explaining the output. Whereas the total loss converges to a value around 3.6 for both x_2 and x_3, the value for x_1 ends at just under 1.

4. Based on your training observations, describe the impact of the different learning rates on the final loss and number of training iterations.

In the case of all three variables, increasing the learning rate purely decreased the time to converge. The number of iterations needed to converge decreased sharply with increases in the learning rate and no underdamped (jumping) response was observed in the range from 0.01 to 0.1 with any of the variables.

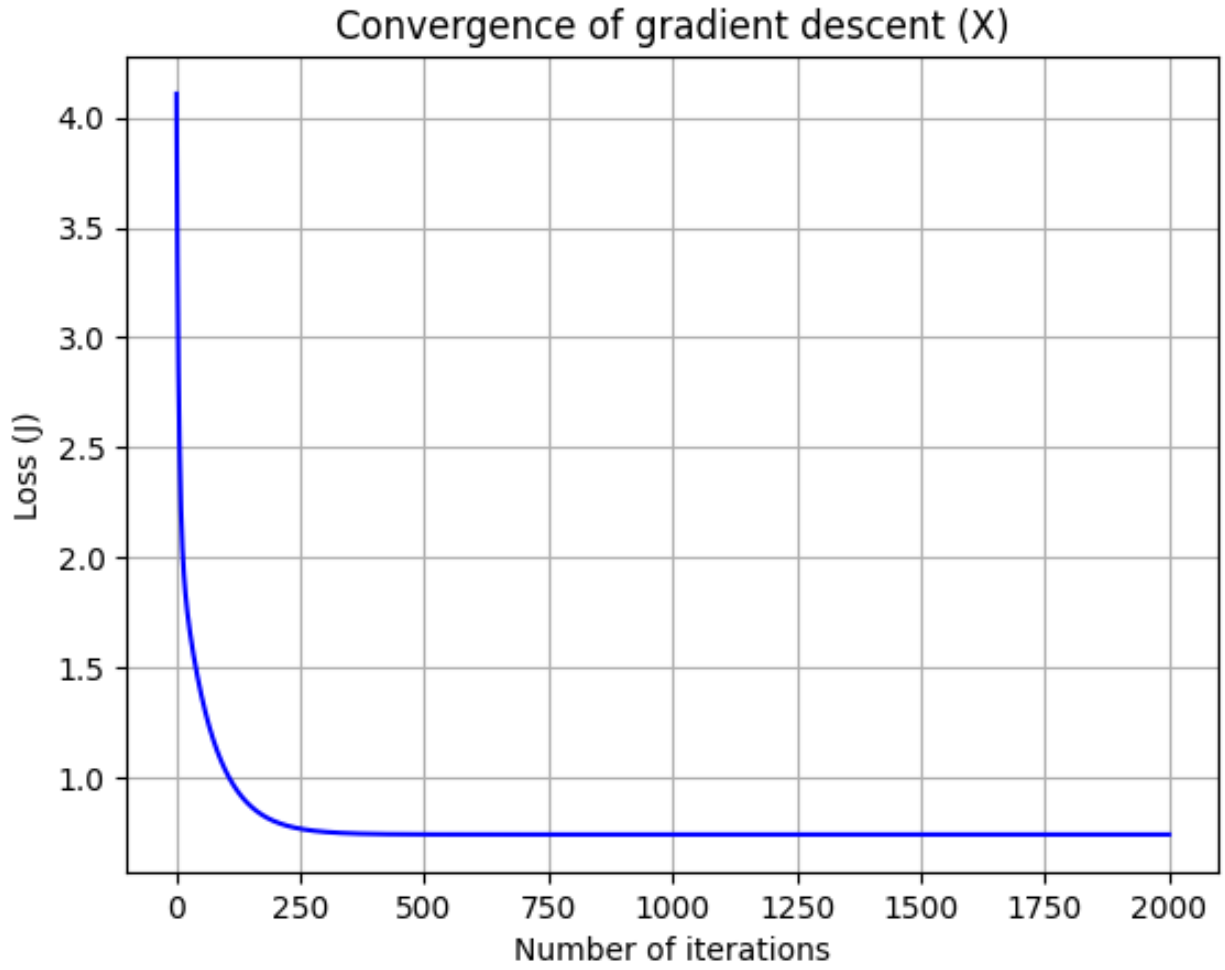
Problem 2: Multi-variable linear regression gradient descent

1. Report the final linear model you found the best.

$\alpha = 0.08$, N = 2000

$$h(x_1, x_2, x_3) = -2.00371919x_1 + 0.53256343x_2 - 0.26560178x_3 + 5.3141666$$

2. Plot loss over the iteration.



3. Based on your training observations, describe the impact of the different learning rates on the final loss and number of training iterations.

Similar to the single-variable problem, higher learning rates cause the model to converge quicker, taking fewer iterations to reach the target value. My example model used a learning rate of 0.08 which produced the model values specified above in 2000 iterations. Over the same number of training epochs a learning rate of 0.01 would not fully converge and would therefore underfit, while the faster learning rate of 0.1 over that same time would fit more tightly. Examples of these scenarios are included in the Jupyter notebook.

4. Predict the value of y for new (x_1, x_2, x_3) values (1, 1, 1), (2, 0, 4) and (3, 2, 1).

$$h(1, 1, 1) = 3.5774090589936436$$

$$h(2,0,4) = 0.24432109722649975$$

$$h(3,2,1) = 0.10253411574010246$$