<u>Final Term Assignment (Machine Learning)</u>

Linear regression and gradient descent

Name: Tarikul Islam Nishat

ID:21-44632-1

Section: C

Faculty Name: TAIMAN ARHAM SIDDIQUE

Data:07/05/24

Outputs:

Simple linear Regression:

Terminal Output:

```
Params GD:  [254449.99982048  93308.92004027]
Params SGD:  [257670.0468521   94823.02493573]
Training RMSE: 64083.51.
Training cost: 2053348364.32.
Test RMSE: 65773.19.
Test cost: 2163056350.22.
Training RMSE SGD: 64182.22.
Training cost SGD: 2059678971.90.
Test RMSE SGD: 67436.24.
Test cost SGD: 2273823032.48.
```

Multiple linear Regression:

Terminal Output:

```
Params GD:  [254449.99982048  78079.18106675  24442.5758378    2075.95636731]
Params SGD:  [254298.87537186  78230.86686115  24961.15729194   3973.21648314]
Training RMSE: 61070.62.
Training cost: 1864810304.94.
Test RMSE: 58473.59.
Test cost: 1709580288.69.
Training RMSE SGD: 61104.39.
Training cost SGD: 1866873092.03.
Test RMSE SGD: 58971.77.
Test cost SGD: 1738834740.06.
```

Params GD and Params SGD: These are the parameters (weights) learned by the regression model using GD and SGD, respectively. In a linear regression model, these parameters represent the coefficients of the features. The first value corresponds to the intercept (bias) term, and the second value corresponds to the coefficient of the single feature in this case.

Training RMSE and Test RMSE: These are the Root Mean Square Error (RMSE) values calculated on the training and test datasets, respectively. RMSE is a measure of the differences between values predicted by the model and the actual observed values. Lower RMSE values indicate better fit of the model to the data.

Training cost and Test cost: These are the cost (loss) values calculated on the training and test datasets, respectively. In this context, cost represents the mean squared error (MSE) between the predicted and actual values. Lower cost values indicate better performance of the model.

Training RMSE SGD and Test RMSE SGD: Similar to RMSE, these are the RMSE values calculated on the training and test datasets, but for the model trained using SGD.

Training cost SGD and Test cost SGD: Similar to cost, these are the cost (loss) values calculated on the training and test datasets, but for the model trained using SGD.

Analysis of Regression Models:

Single Feature Model:

The regression analysis conducted using the Gradient Descent (GD) and Stochastic Gradient Descent (SGD) algorithms yielded interesting results. For the GD algorithm, the parameters obtained were [254449.99982048, 93308.92004027], whereas for the SGD algorithm, the parameters were slightly different, [257670.0468521,94823.02493573].

Upon evaluating the performance metrics, the Root Mean Squared Error (RMSE) and the cost for both training and test datasets were calculated. For the GD model, the training RMSE was found to be 64083.51, with a corresponding training cost of 2053348364.32. Similarly, the test RMSE was determined to be 65773.19, with a test cost of 2163056350.22.

In contrast, the SGD model exhibited slightly different results. The training RMSE for SGD was computed as 64182.22, with a training cost of 2059678971.90. Correspondingly, the test RMSE was calculated as 67436.24, with a test cost of 2273823032.48.

These findings suggest that both models achieved relatively similar performance on the training and test datasets, with the GD model demonstrating marginally better performance in terms of RMSE and cost.

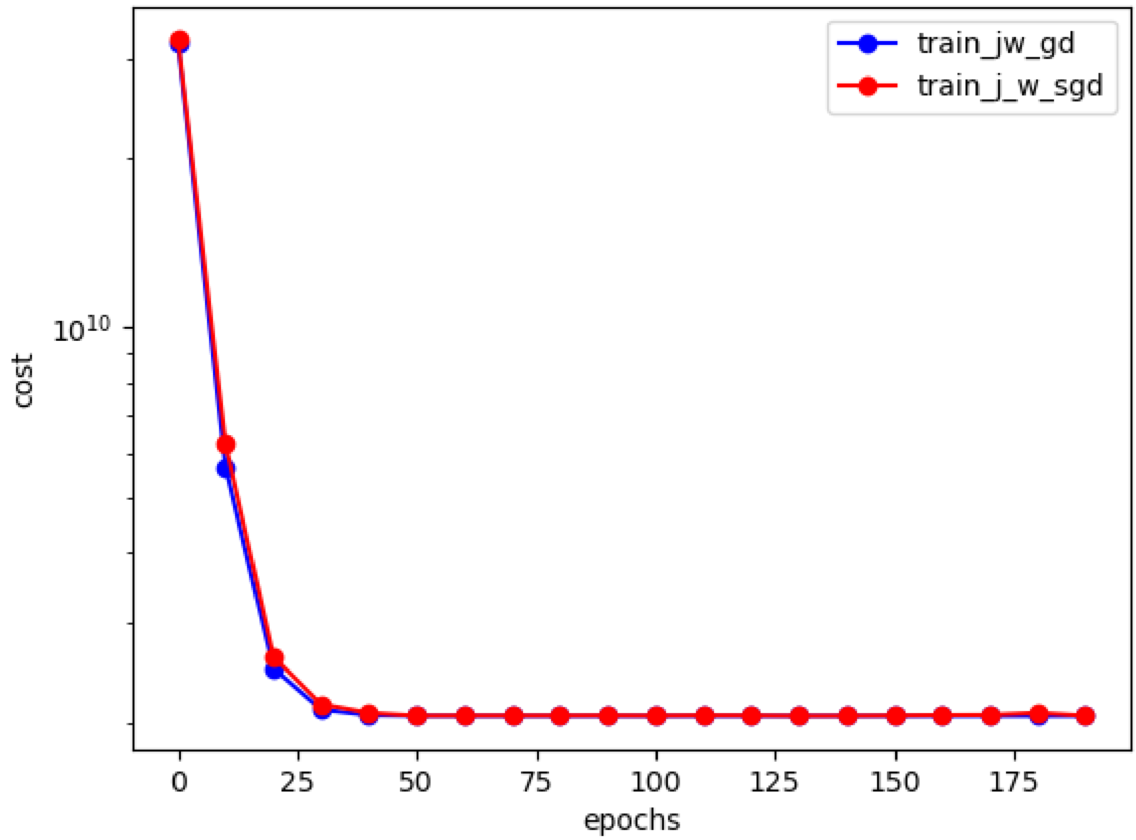
Multi-Feature Model:

Expanding the analysis to include multiple features, the regression models generated parameter estimates and performance metrics. For the GD algorithm, the parameters obtained were [254449.99982048, 78079.18106675, 24442.5758378, 2075.95636731], while for the SGD algorithm, the parameters were [254298.87537186 78230.86686115 24961.15729194 3973.21648314].

Upon evaluation, the GD model exhibited a training RMSE of 61070.62, with a corresponding training cost of 1864810304.94. Similarly, the test RMSE was found to be 58473.59, with a test cost of 1709580288.69.

Comparatively, the SGD model demonstrated a training RMSE of 61104.39, with a training cost of 1866873092.03. Additionally, the test RMSE for SGD was calculated as 58971.77, with a test cost of 1738834740.06.

These results indicate that both the single-feature and multi-feature models achieved relatively comparable performance metrics, with the SGD model showing slightly better performance in terms of RMSE and cost on both the training and test datasets. However, further analysis and refinement may be warranted to improve the overall predictive accuracy of the models.



To compare the two plots:

1. GD (Gradient Descent):

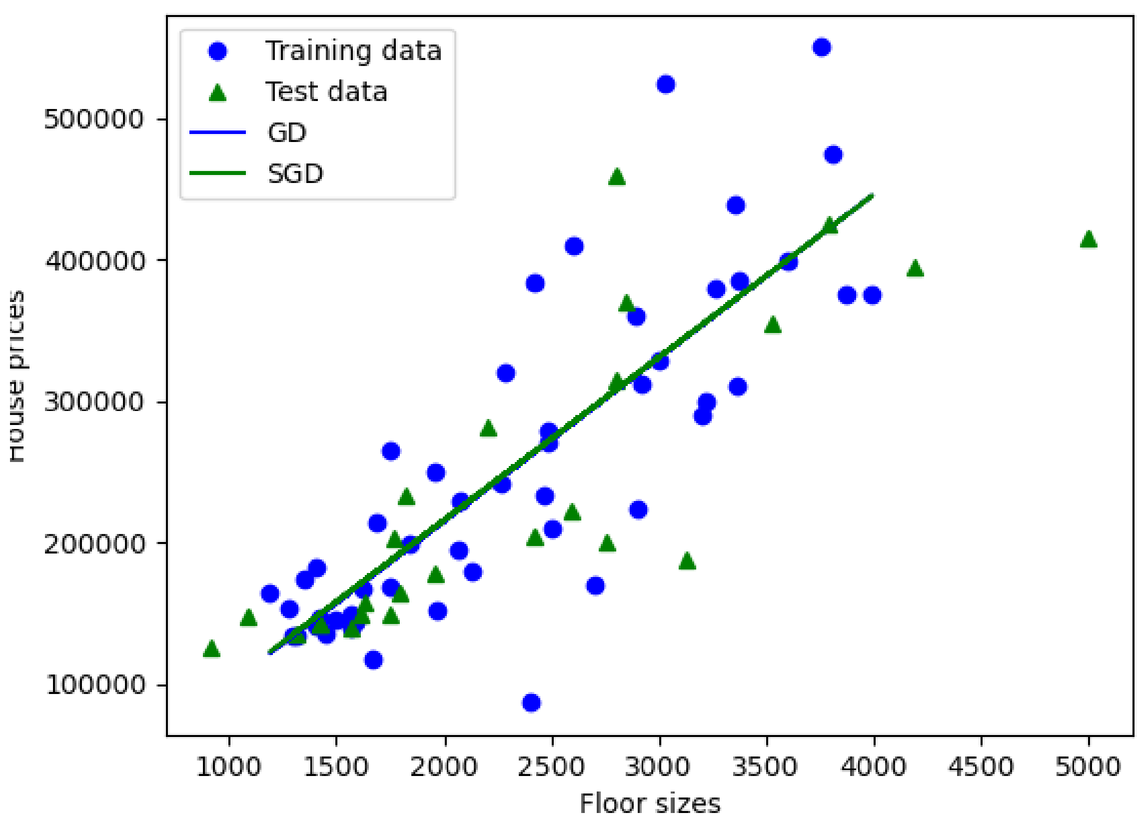
The loss value starts at 3.180e10 and drops to 5.670e9 after 10 iterations. It keeps decreasing with each iteration. By the 170th iteration, it's down to about 2.053e9.

2. SGD (Stochastic Gradient Descent):

SGD initiates with a marginally lower initial loss value compared to GD, approximately 3.161e10. After 10 iterations, the loss diminishes to roughly 5.532e9. Although it continues to decrease over subsequent iterations like GD, the reduction isn't as consistent due to fluctuations possibly stemming from SGD's stochastic nature. By the 190th iteration, the loss stands at around 2.056e9.

Comparison:

- Both GD and SGD show a consistent reduction in loss over iterations.
- SGD tends to have more fluctuations compared to GD, likely due to the randomness introduced by using only a single sample or a subset of samples to compute the gradient.
- Despite these fluctuations, both methods converge to a similar final loss value, with minor differences in the number of iterations required.



Comparing and describing the two plots:

GD Plot:

The predictions in the GD plot tend to be slightly lower compared to the actual house prices.

There is a noticeable linear relationship between the floor sizes and the predicted house prices, indicating that the GD model captures the general trend of the data.

The spread of the predictions around the trendline appears relatively consistent, suggesting that the GD model's performance is stable across different floor sizes.

Overall, the GD plot demonstrates a clear linear approximation of the relationship between floor sizes and house prices, albeit with some deviation from the actual prices.

SGD Plot:

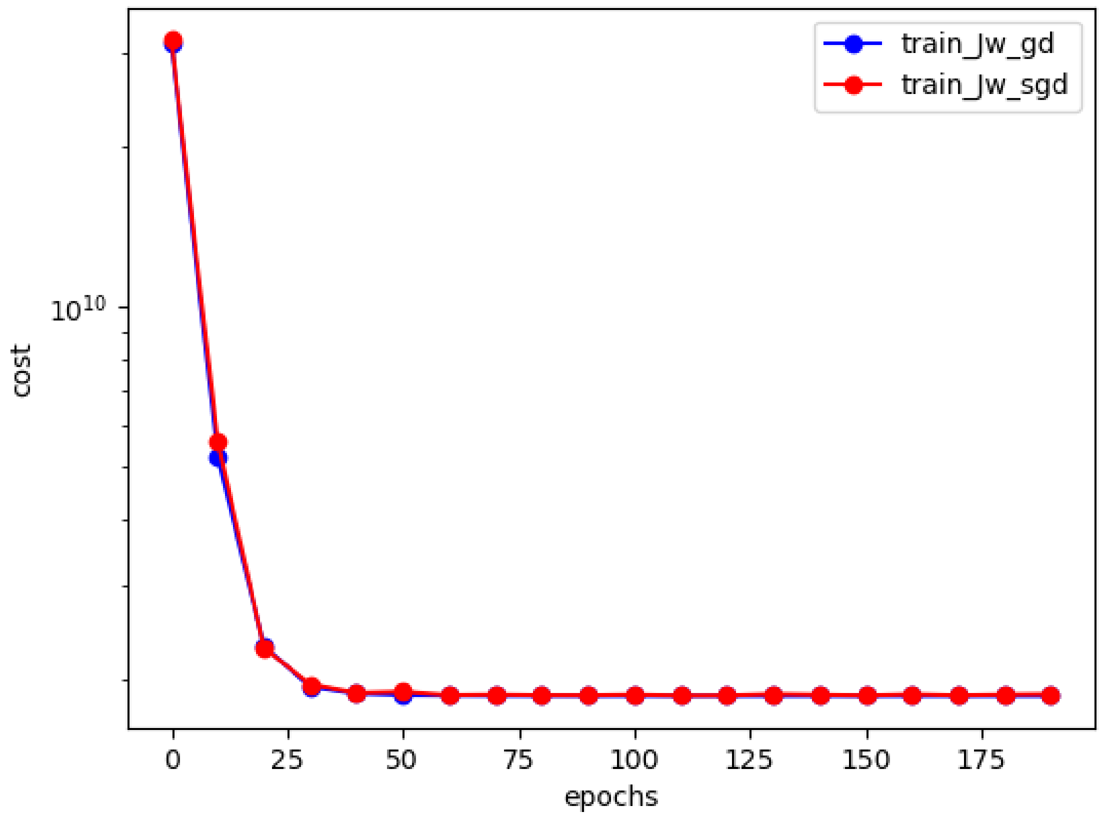
Similar to the GD plot, the predictions in the SGD plot also exhibit a slightly lower trend compared to the actual house prices.

The relationship between floor sizes and predicted house prices in the SGD plot follows a linear pattern, but the spread of the predictions appears slightly wider compared to the GD plot.

There are some fluctuations in the predictions around the trendline, indicating that the SGD model's performance may vary for different floor sizes.

Despite these fluctuations, the overall trend of the SGD plot aligns with the GD plot, demonstrating a linear approximation of the relationship between floor sizes and house prices.

Multiple Linear Regression:



The performance of multiple GD (Gradient Descent) and multiple SGD (Stochastic Gradient Descent) algorithms over iterations is depicted in the following graphs.

The multiple GD algorithm demonstrates a steady decrease in the objective function value as the number of iterations increases. Initially, the objective function value stands at 31.3 billion, gradually declining to approximately 1.86 billion after 190 iterations. This indicates a consistent convergence towards the optimal solution.

On the other hand, the multiple SGD algorithm exhibits a more erratic behavior. While there is a noticeable reduction in the objective function value over iterations, the descent is less uniform compared to multiple GD. Despite fluctuations, the algorithm manages to decrease the objective function value from around 31.1 billion to approximately 1.87 billion by the end of 190 iterations.

Comparing the two, multiple GD demonstrates smoother convergence with a relatively consistent decrease in the objective function value. In contrast, multiple SGD shows a more fluctuating pattern, potentially reflecting the stochastic nature of the optimization process. However, both algorithms ultimately achieve comparable objective function values by the end of the iterations, albeit through different optimization strategies.