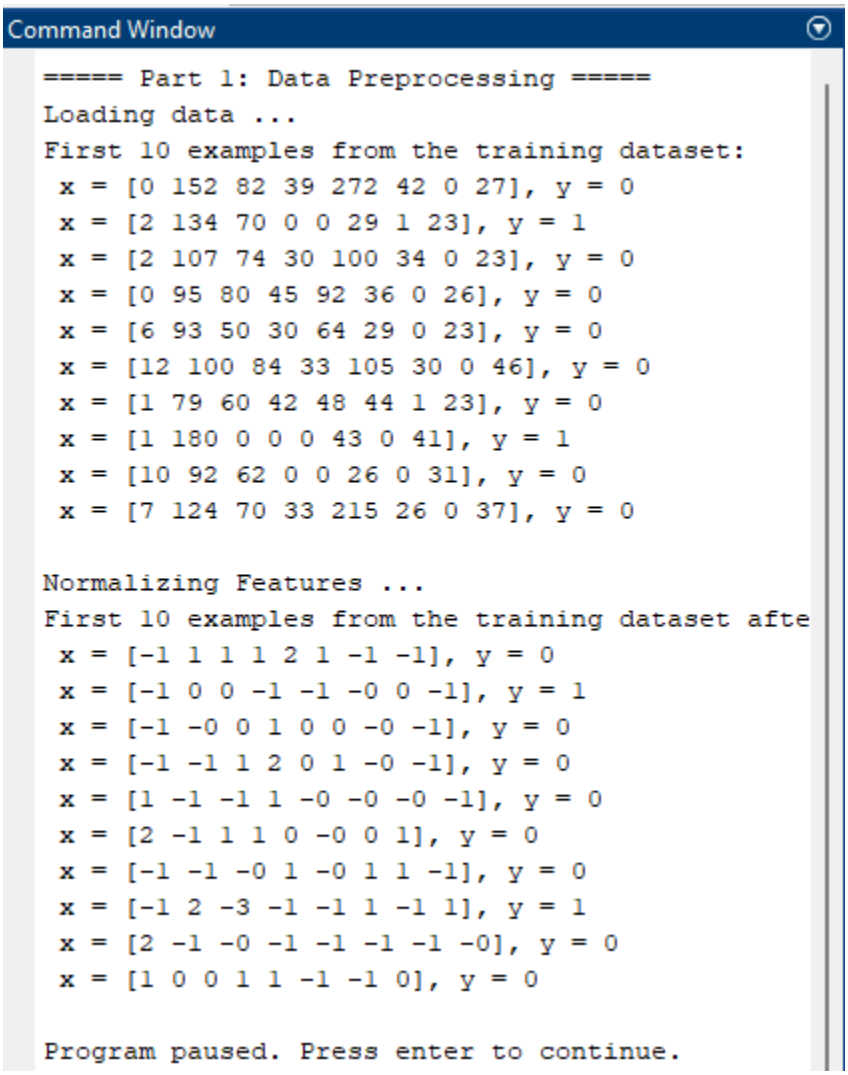
Brandon Nguyen
COMPE510 - Fall 2025
827813045

Programming Assignment 4 - Neural Networks

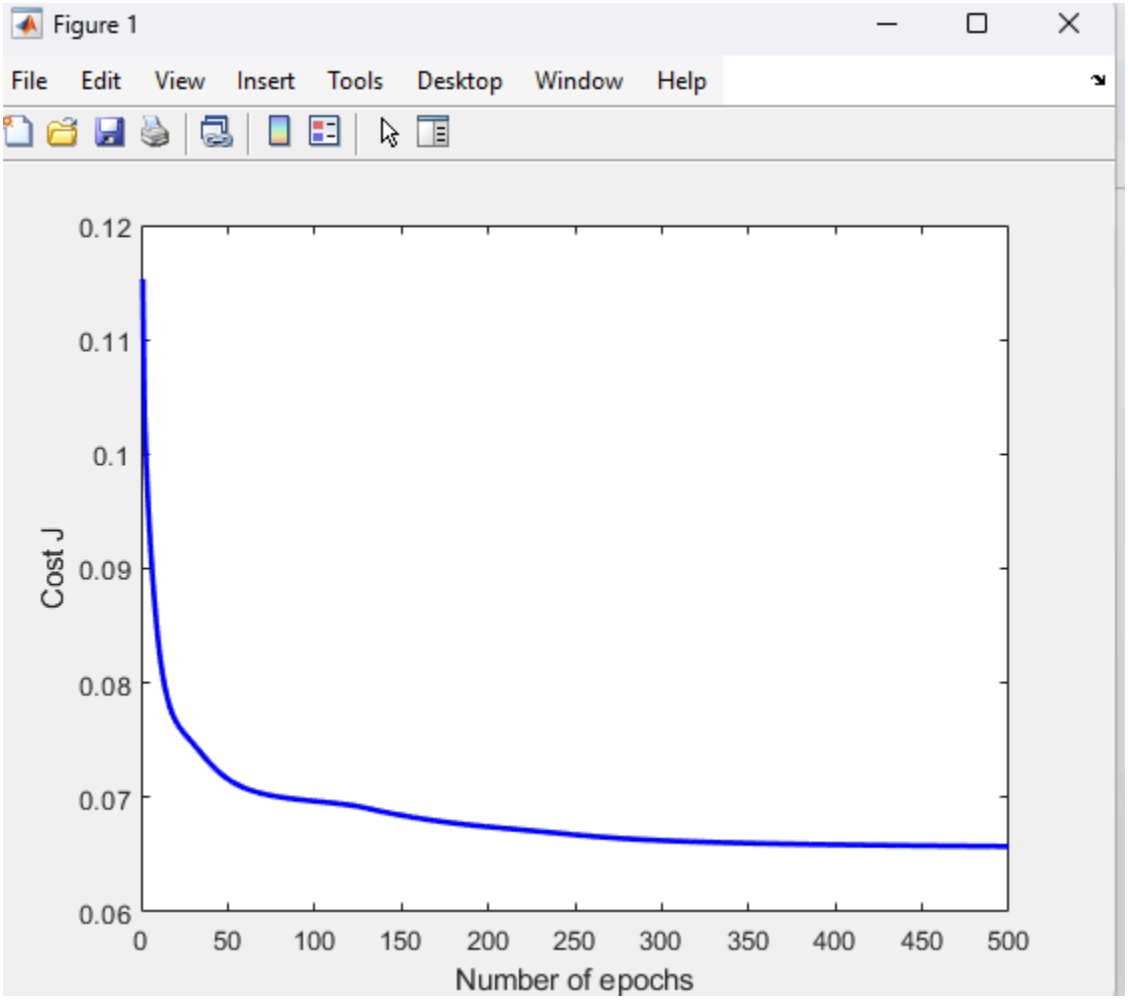
A) Results

- **Data Processing**



- **Train Neural Networks**

```
===== Part 2: Train Neural Network =====

ans =

   537     8


ans =

   537     1

beta1 computed from gradient descent:
 0.944290
 -2.148296
 1.055481
 0.109048
 0.373357
 0.462914
 1.265629
 0.661614
 3.425511
 -0.975980
 0.442228
 -0.179131
 0.771703
 0.100353
 -0.999896
 -1.546272
 1.428559
 -0.791106
 0.927438
 2.206038
 -0.109060
 -1.528203
 2.813660
 1.161504
 -0.926973
 0.013324
 3.298196

beta2 computed from gradient descent:
 -5.585257
 2.943422
 2.971851
 3.225652

Program paused. Press enter to continue.
```



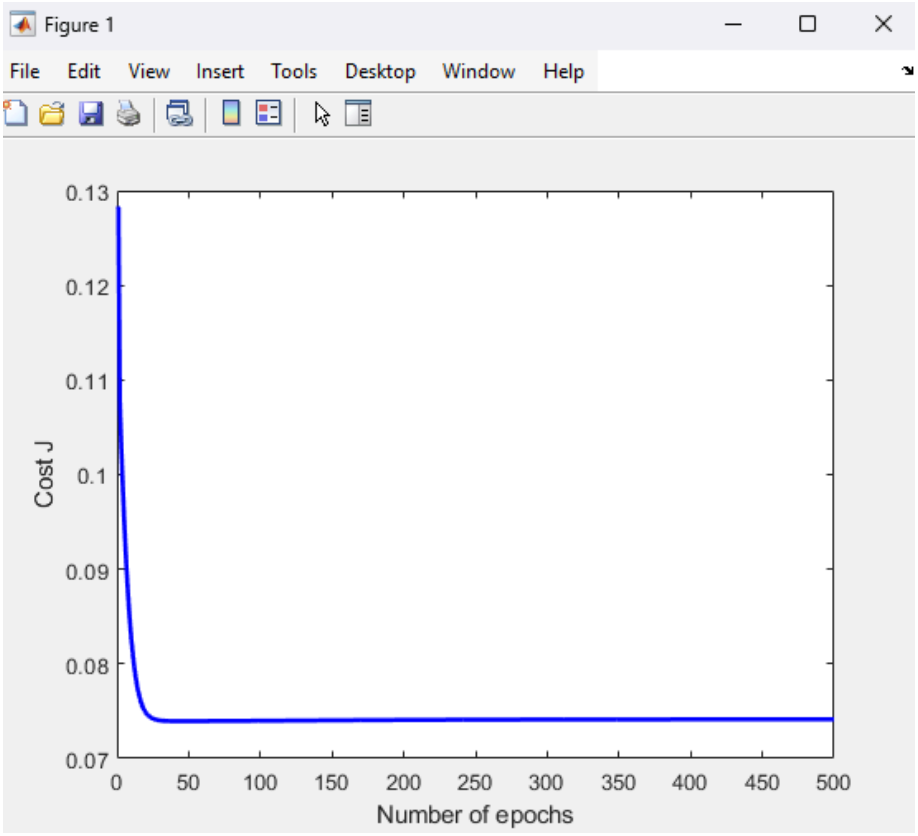
- **Evaluate Performance**

```
Program paused. Press enter to continue.
===== Part 3: Evaluate Performance =====
Accuracy:
 74.025974
```

B) Questions

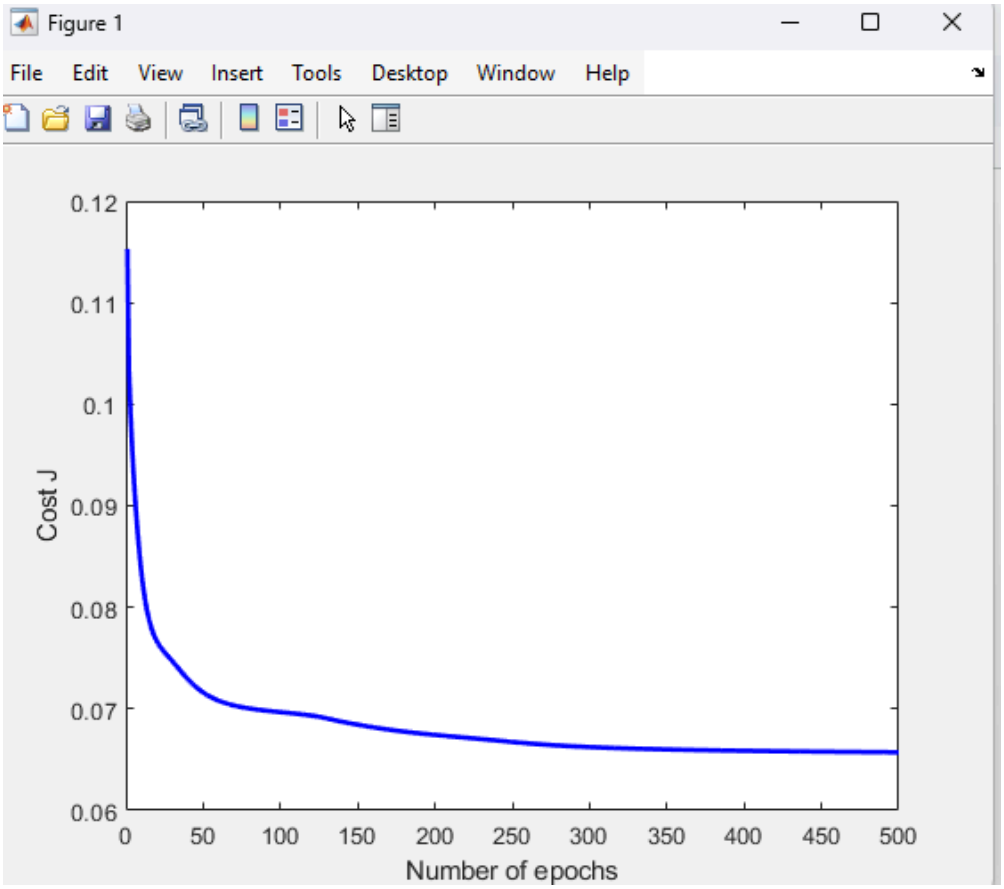
- **Train three neural network models with different numbers of hidden units. For each model, report the accuracy on the test set. Ensure that the training algorithm has fully converged before evaluating performance. Based on your results, analyze how the number of hidden units affects model performance. Which configuration produced the best results for your dataset, and why do you think that is the case?**

Test 1: Hidden Unit = 1



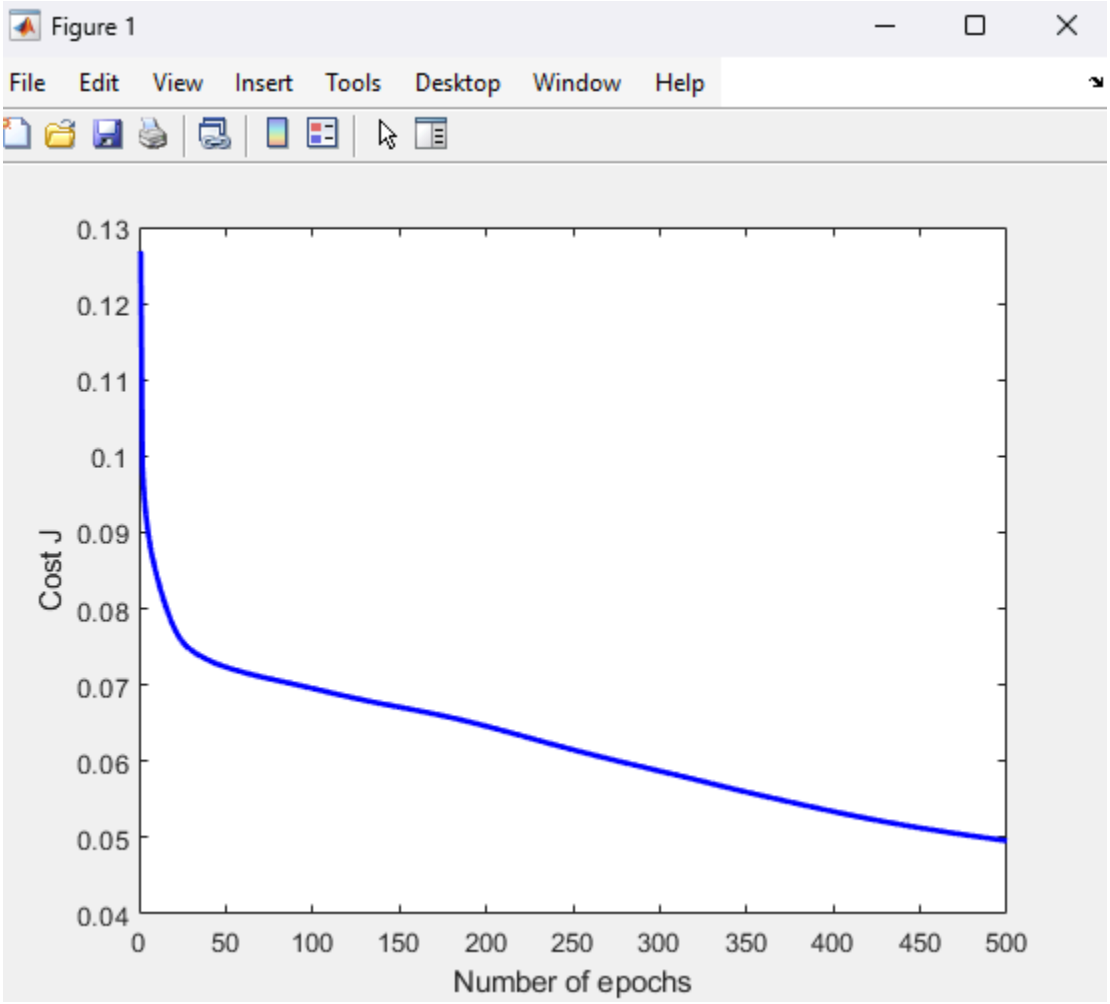

```
Program paused. Press enter to continue.
===== Part 3: Evaluate Performance =====
Accuracy:
 76.190476
```

Test 2: Hidden Unit = 3




```
Program paused. Press enter to continue.
===== Part 3: Evaluate Performance =====
Accuracy:
 74.025974
```

Test 3: Hidden Unit = 10




```
Program paused. Press enter to continue.
===== Part 3: Evaluate Performance =====
Accuracy:
 68.831169
```

After training three neural network models with different sets of numbers, it is clear that setting the hidden units variable to a number like 3 is the best fit because its convergence is the best fitting. In comparison to the other two, the graph for hidden units = 1 is underfitting because it has a bad convergence, dropping sharply before flatlining. Additionally, the graph for hidden units = 10 is also not as good because it

overfits and has a slow convergence. Now when we look at the accuracy of these three graphs, we would only pay attention to hidden units 1 and 3 because they have the greater values of accuracy. Although hidden units 1 has a higher accuracy than hidden units 3, the graph for hidden units = 3 has the best convergence, clearly showing that it is the best test value for the dataset and model. So one conclusion we can come to is that a smaller hidden unit value would be good for this dataset and model in general.

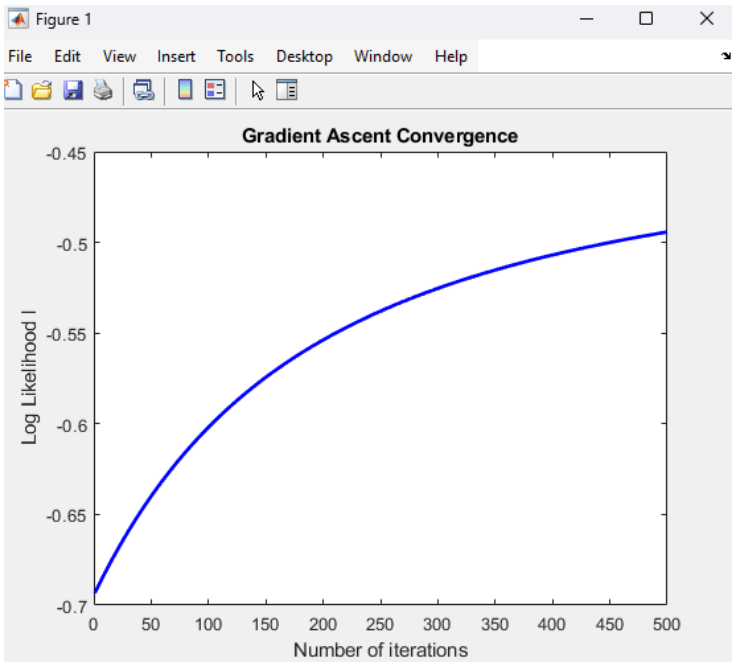
- **Based on your training and test accuracy, do you think your model is overfitting, underfitting, or generalizing well?**

According to the training and test accuracy of a couple different neural network models, I believe that this specific model is generalizing well because it is accurately showing differences based on different values being tested or trained, and the accuracies for the three tests that I conducted are all relative high, showing good training for this neural networks model.

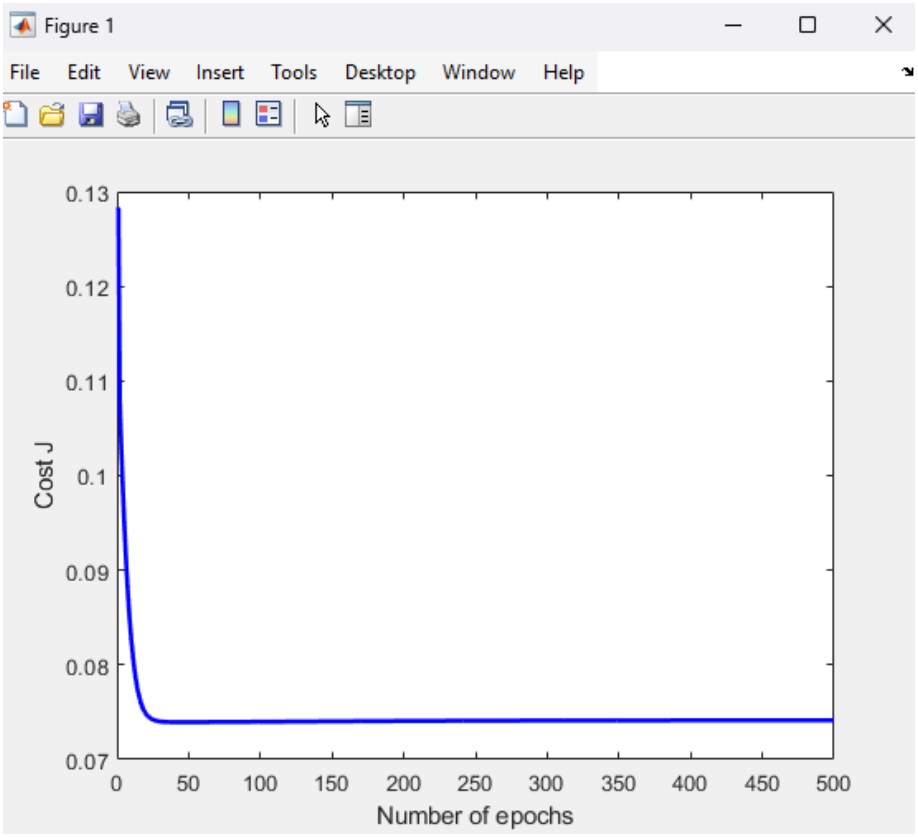
- **Compare the performance of the logistic regression model and the neural network model you have trained in terms of test accuracy, convergence behavior, and prediction confidence. Which model performed better, and why do you think that is? What does this tell you about the representational power and limitations of each model?**

Logistic Regression Model:

```
Program paused. Press enter to continue.
===== Part 3: Evaluate Performance =====
Accuracy:
 76.190476

Predicted output:
 0
>>
```

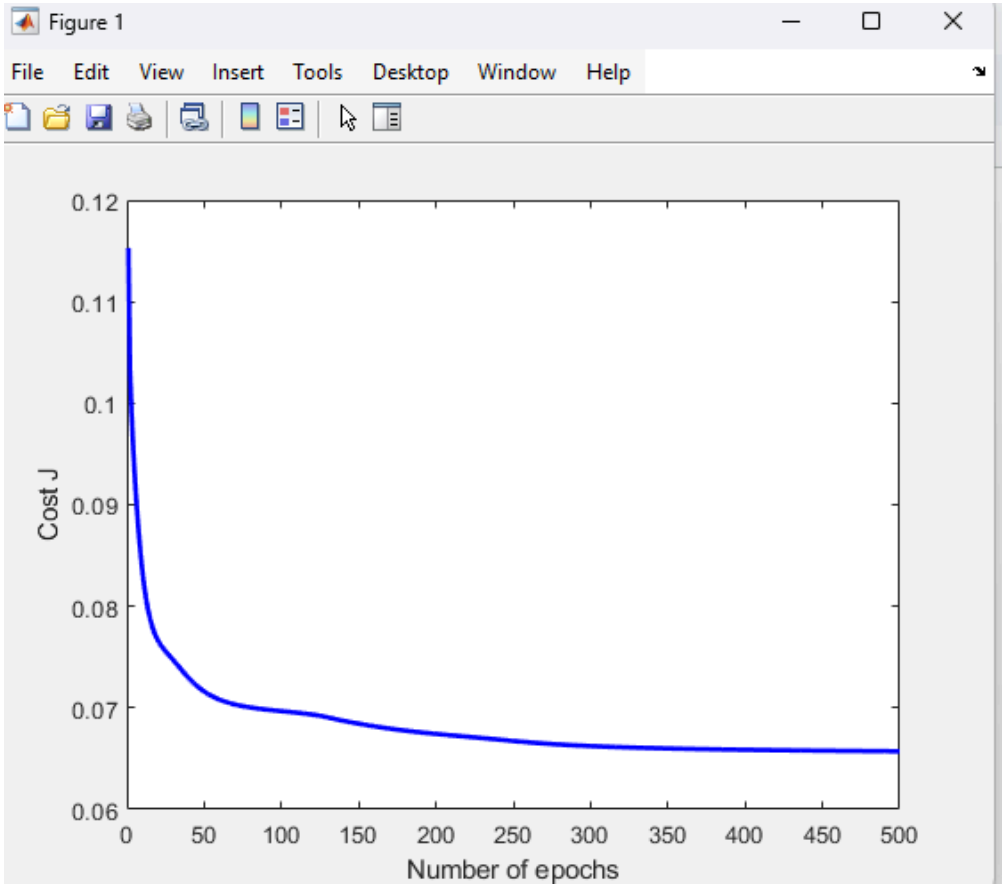


Neural Network Model:
Hidden Unit = 1




```
Program paused. Press enter to continue.
===== Part 3: Evaluate Performance =====
Accuracy:
 76.190476
```

Hidden Unit = 3




```
Program paused. Press enter to continue.
===== Part 3: Evaluate Performance =====
Accuracy:
 74.025974
```

When comparing the performance between the logistic regression model and the neural network model based on test accuracy, convergence, behavior, and prediction confidence, I believe that the logistic regression model is better when working with the Pima Indians Diabetes dataset. This is because the accuracy from training the model is already high off of one test. Additionally, using the conclusion that a lower hidden units value is better for neural network models, we will be comparing the logistic regression model to the neural network models where the hidden units are set to 1 and 3. Since we said that setting the hidden units to 3 is the best test, it actually had a lower accuracy than the logistic regression model. After changing the hidden units value to 1 was when the accuracy finally matched with the logistic regression model. In conclusion, although

it depends on what specific dataset is being used, in this case the accuracies and performance evaluation support the idea that logistic regression is the better model, especially for working with the Pima Indians Diabetes dataset. Furthermore, the neural networks model required some adjusting with test values in order to be on par with the logistic regression model for analyzing this dataset.

C) Summary

In this fourth programming assignment, I believe that the implementation of each of the program files went well in connection to the main program file. The things that went well regarding this fourth assignment include training the neural network model and producing outputs from various testing and training values. On the other hand, some challenges along the way included debugging my code when it did not produce an output at all, and working with the multiplication matrix in part 2 of the main module code despite getting it to work still according to the output pictures. Overall, this fourth assignment was efficient in teaching me about new topics like neural networks, training neural network models, backpropagation algorithm, overfitting, and more.