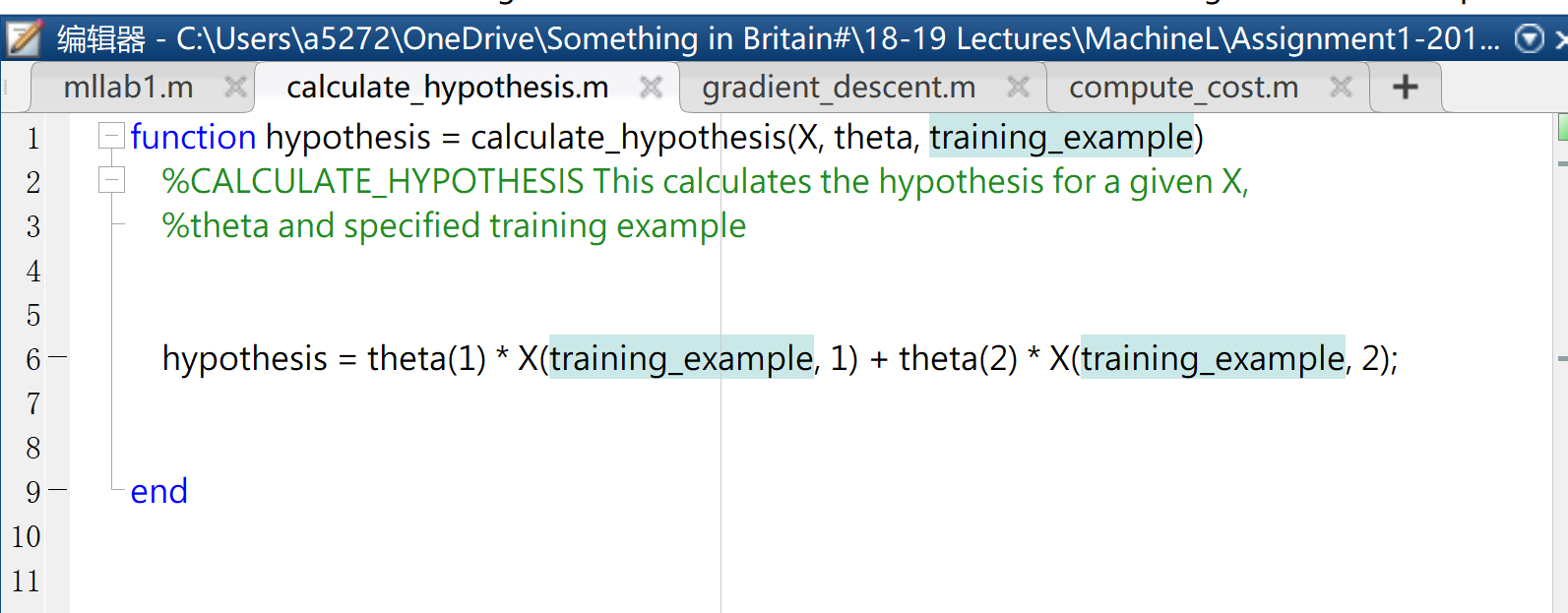
# Ass1 (part1) 4 MachineL

# 1.Linear regression

a)

Modify the function calculate\_hypothesis.m to return the predicted value for a single specified training example. Include in the report the corresponding lines from your code.

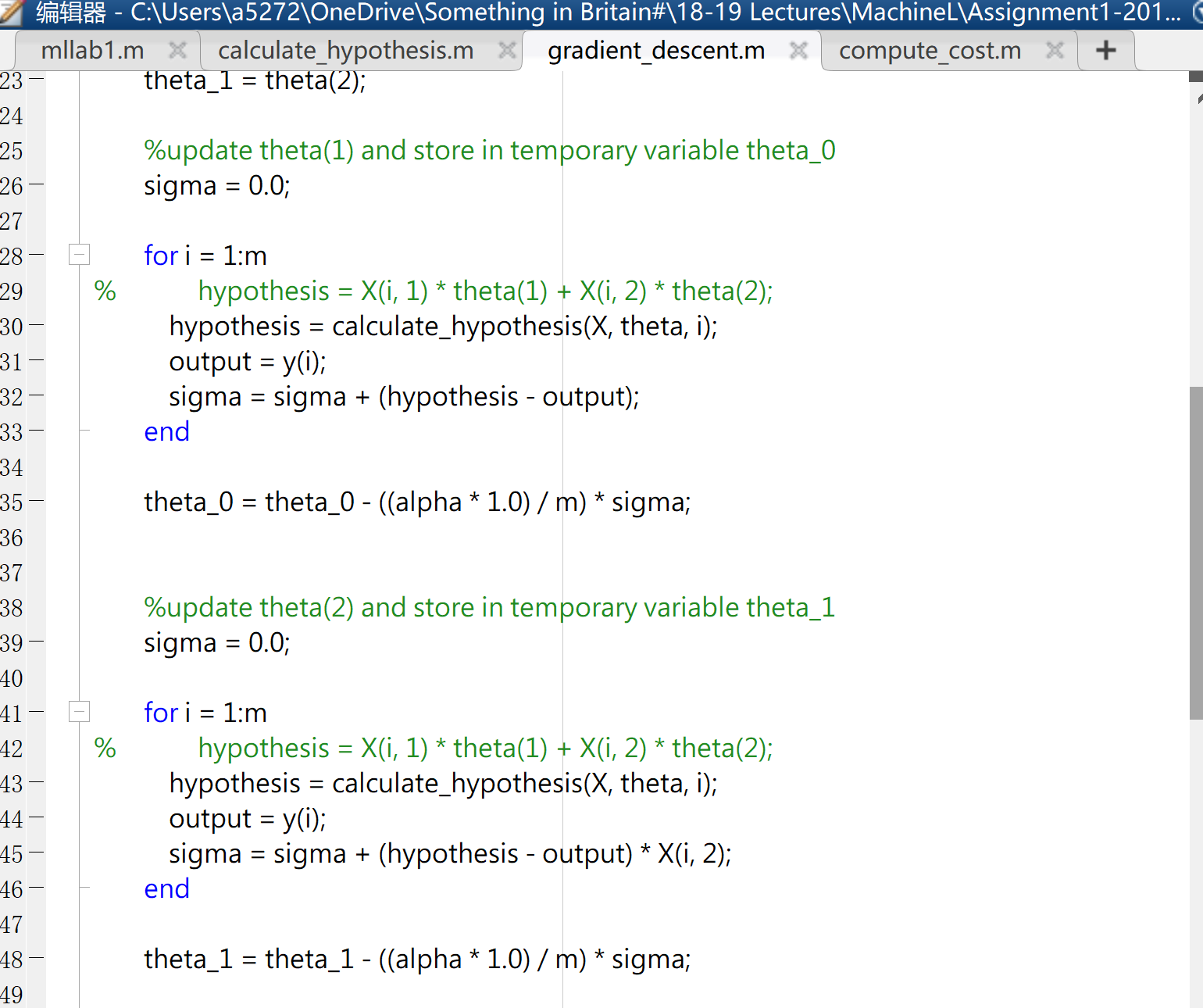
[5 points]



b)

Modify it to use the calculate\_hypothesis function. Include the corresponding lines of the code in your report.

[5 points]

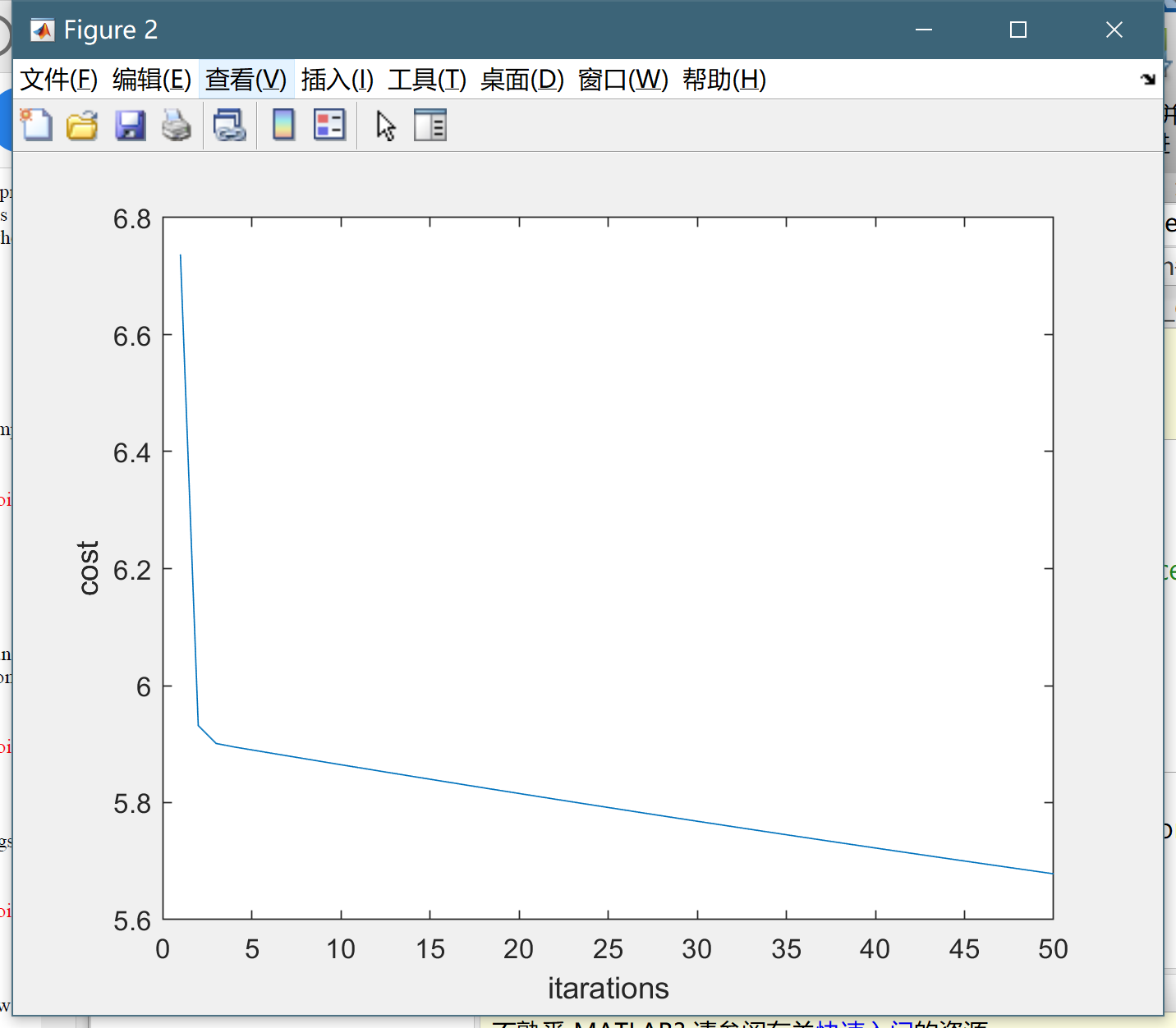


c)

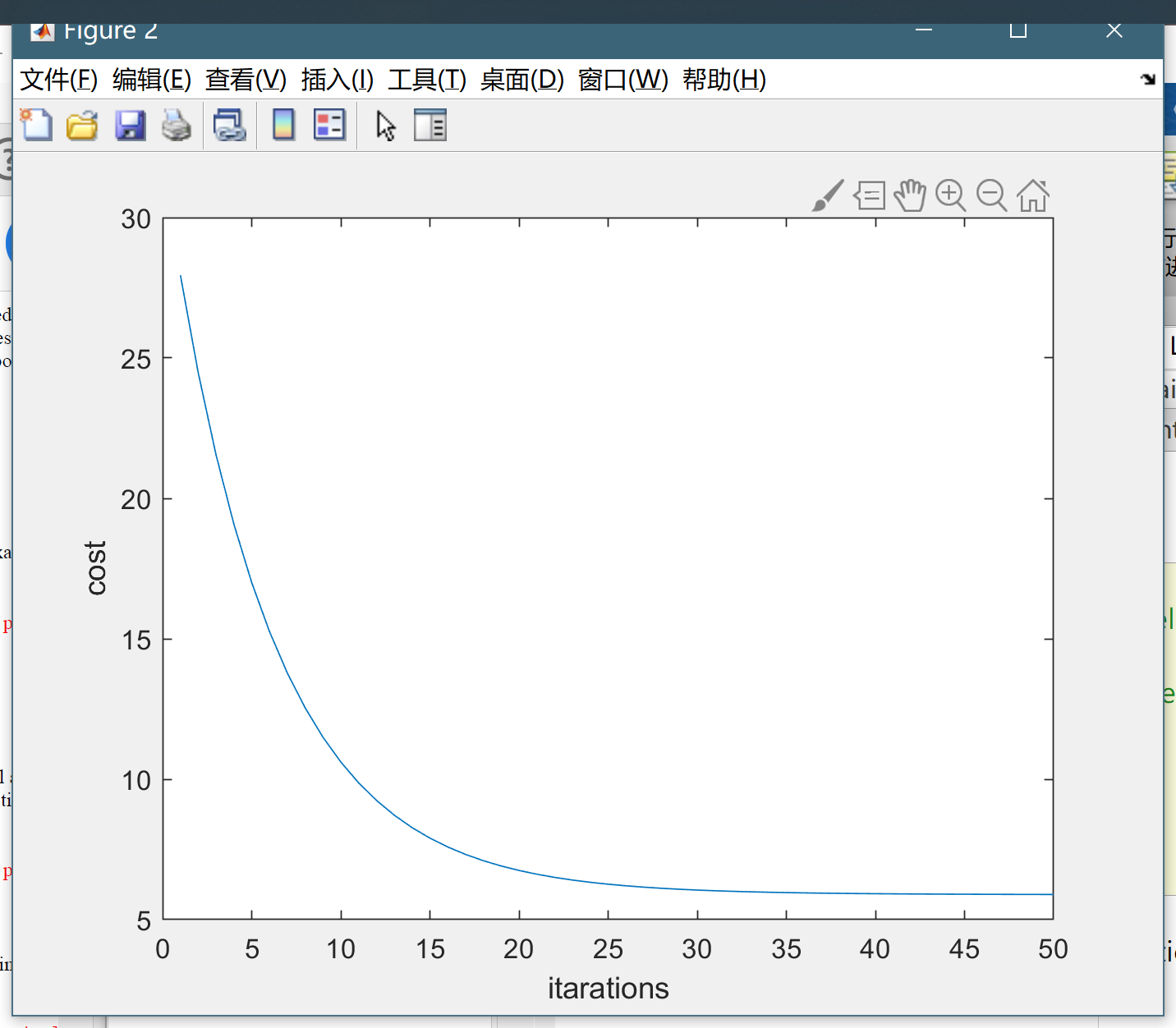
Observe what happens when you use a very high or very low learning rate. Document and comment on your findings in the report.

[5 points]

Hi-learning rate (0.01):



Low-Learning Rate (0.001):



When using a lower learning rate, the curve we got is smoother and it is more likely to reach the local optimum.

# 2.Linear regression with Multiple Variables

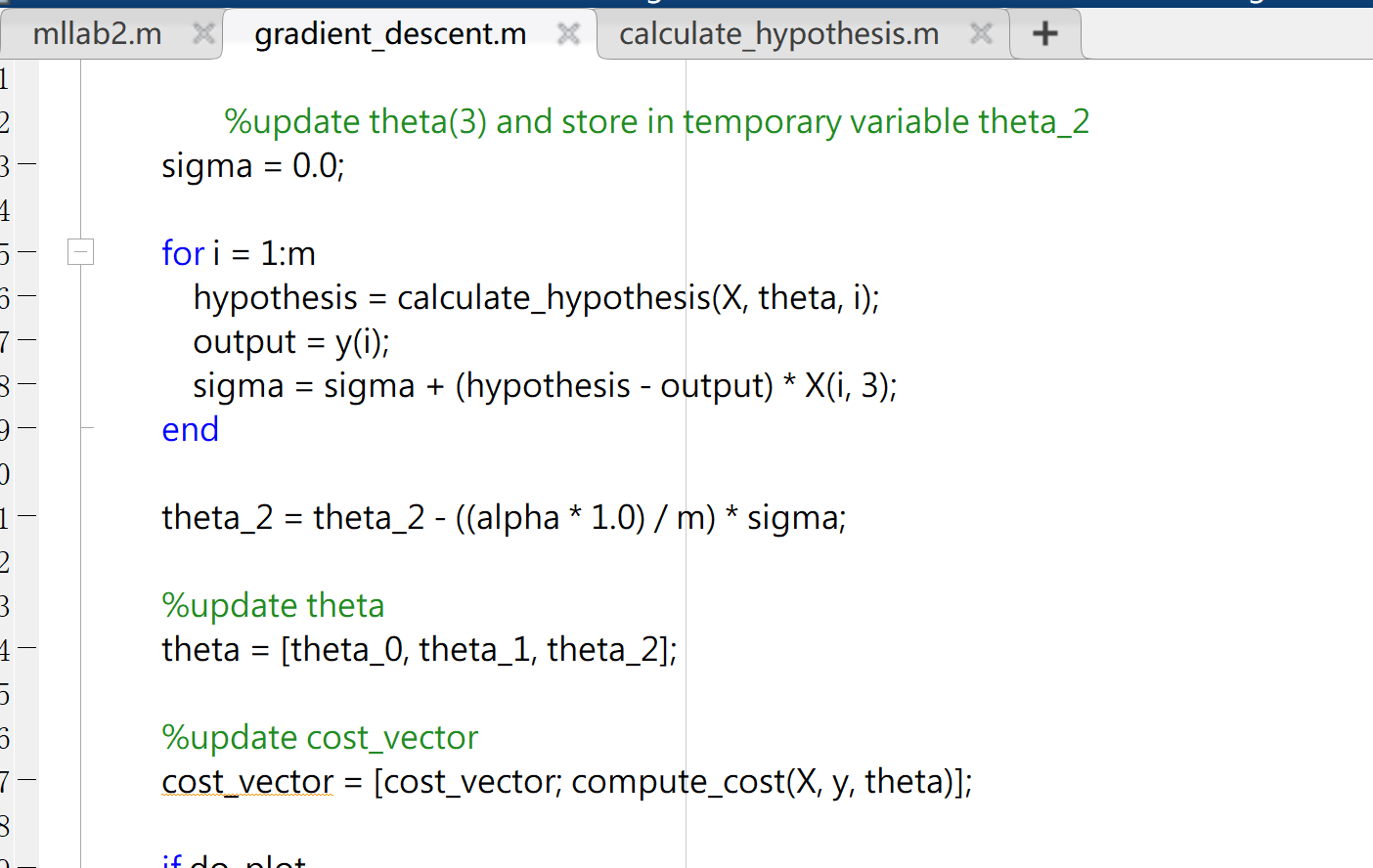
a)

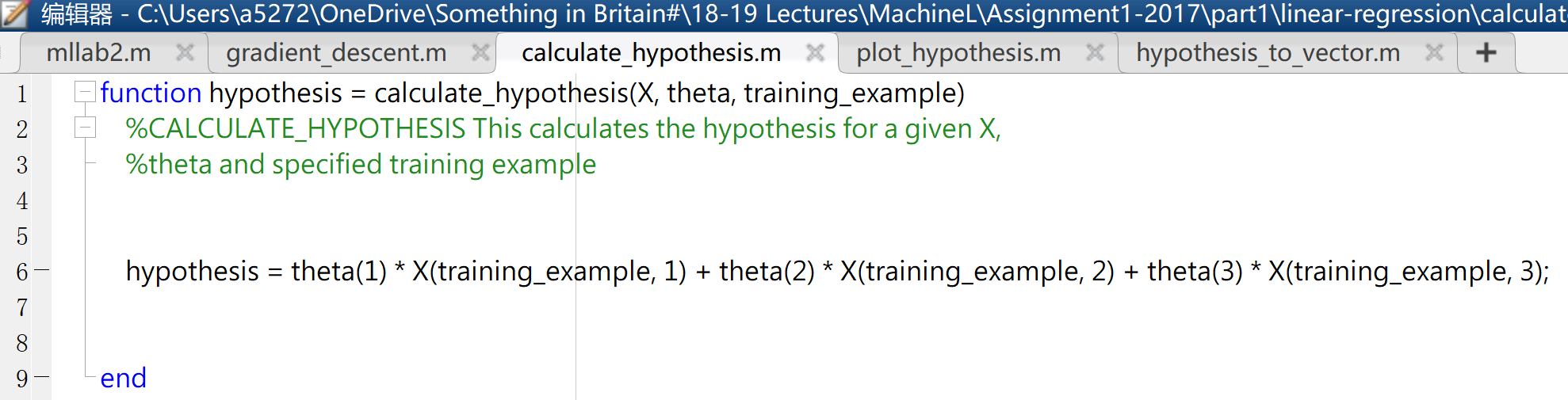
Modify the functions calculate\_hypothesis and gradient\_descent to support the new hypothesis function. This

should be sufficiently general so that we can have any number of extra variables. Include the relevant lines of the code

in your report.

[5 points]



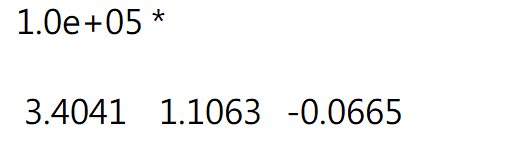
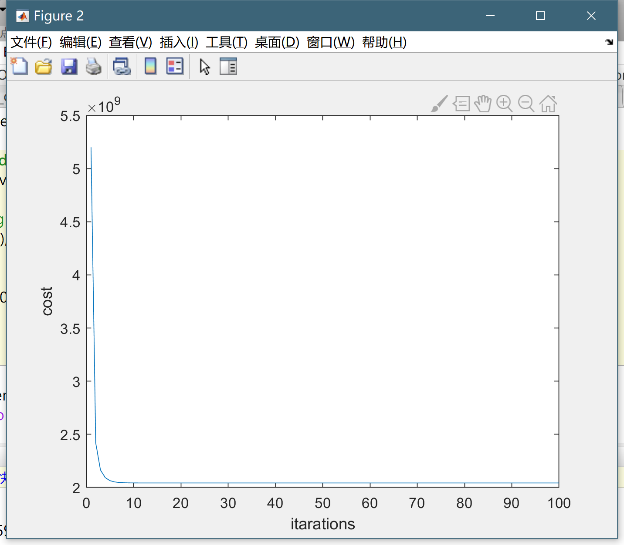


b)

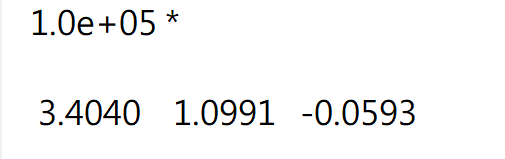
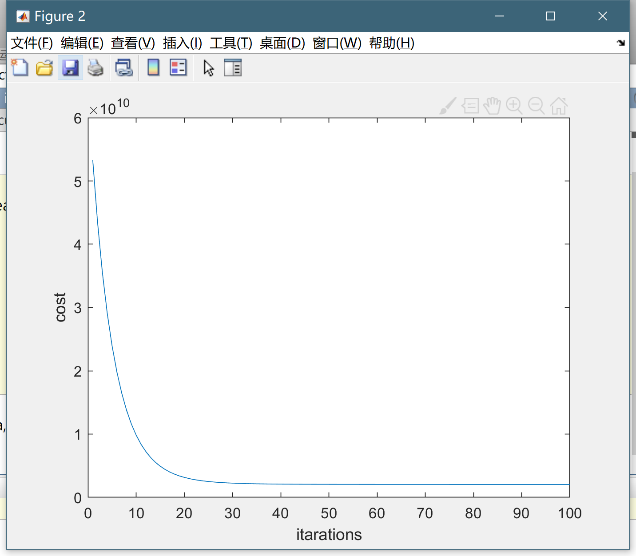
Run mllab2.m and see how different values of alpha affect the convergence of the algorithm. Print the theta values found at the end of the optimization. Include the values of theta and your observations in your report.

[5 points]

α = 0.8 θ value

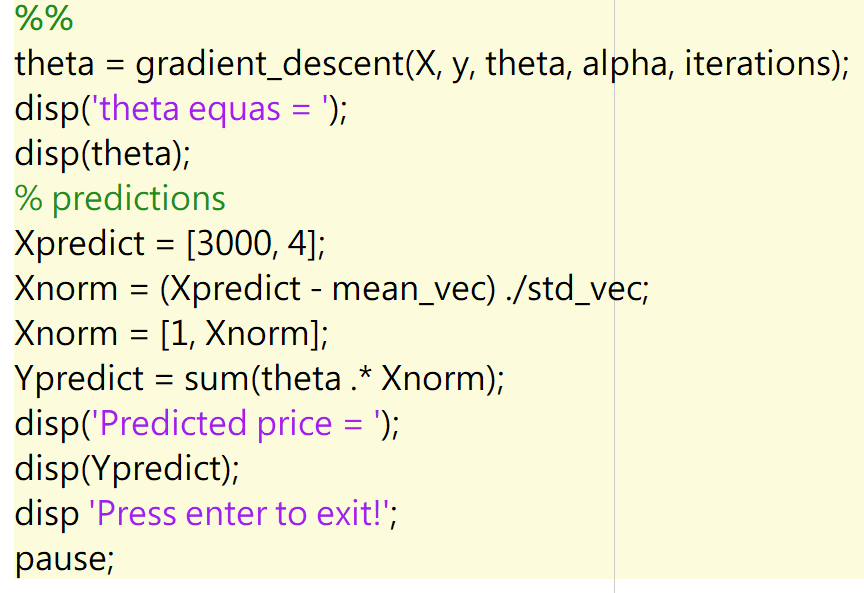


α = 0.1 θ value equal to



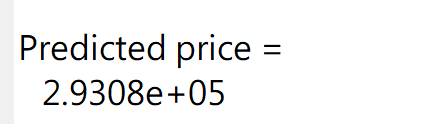
Observations: α has a strong effect to the interaction number. When α is low, converge interaction number is low, which means the speed of convergence is faster.

Modify the function as below



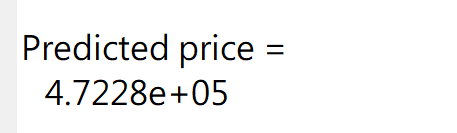
When 1650 sq.ft. 3 bedrooms

The prediction value is



When 3000 sq.ft. 4 bedrooms

The prediction value is



# 3.Regularzed Linear regression

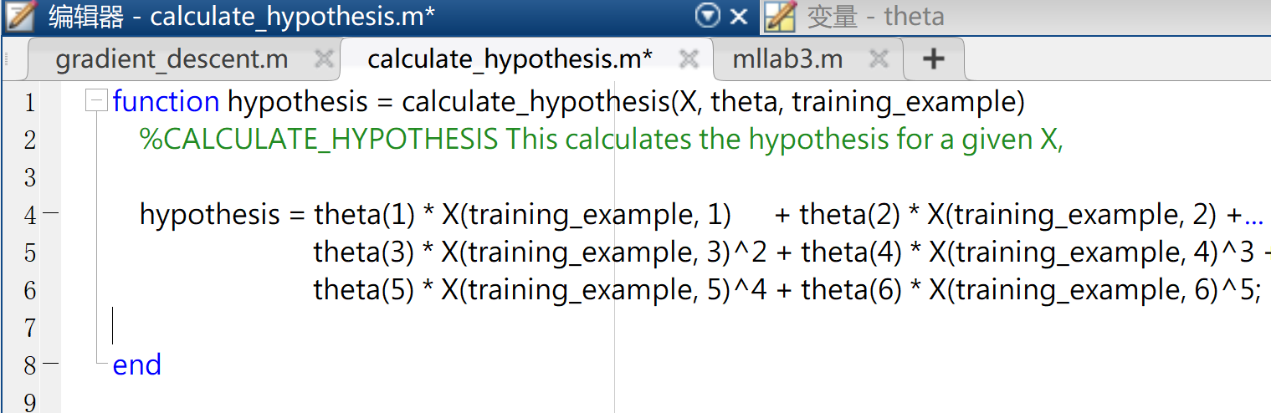
a)

Note that the punishment for having more terms is not applied to the bias. This cost function has been implemented already in the function compute\_cost\_regularised. Modify gradient\_descent to use the compute\_cost\_regularised method instead of compute\_cost. Include the relevant lines of the code in your report and a brief explanation.

[5 points]

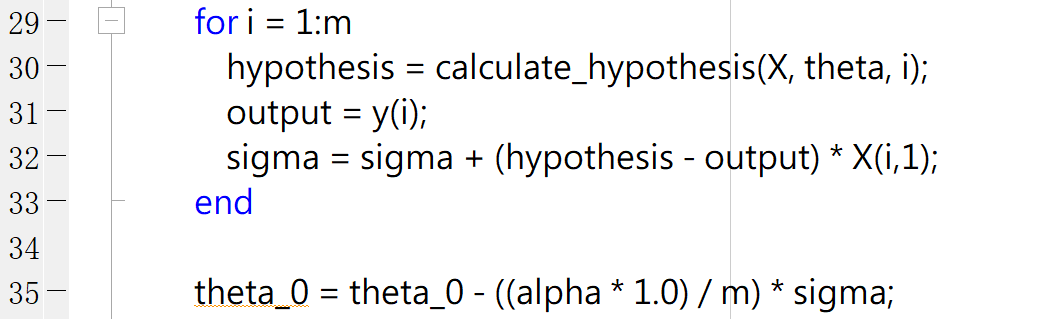


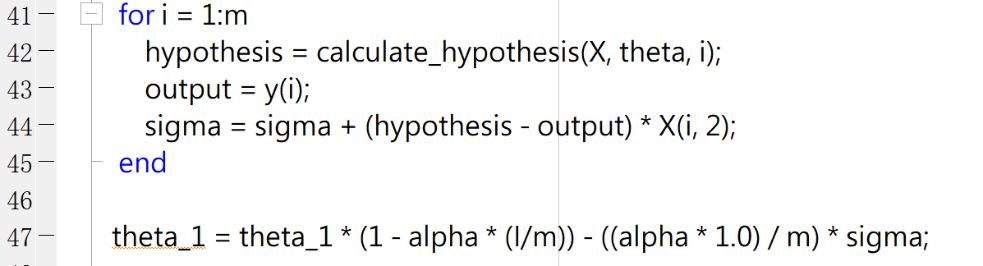




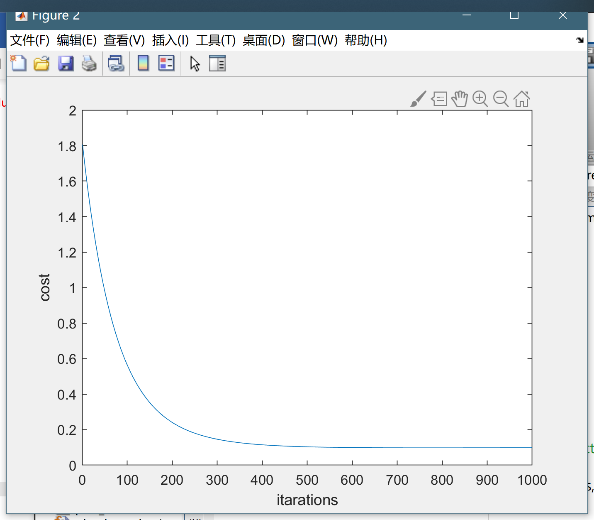
Next, modify *gradient\_descent* to incorporate the new cost function. Include the relevant lines of the code in your report.

[5 points]





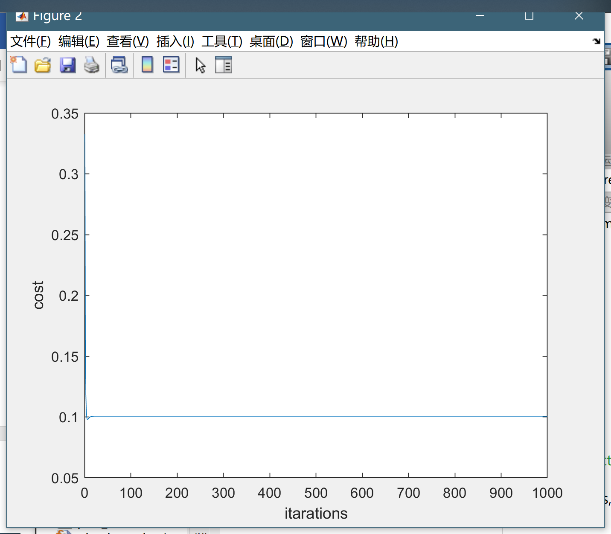
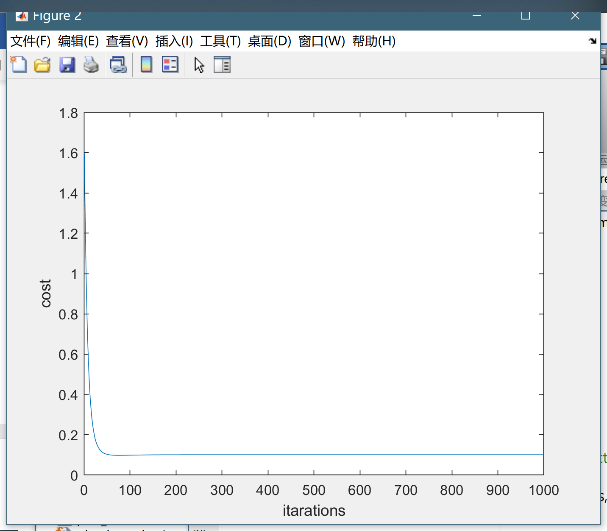
First of all, find the best value of alpha to use in order to optimize best. Report the value of alpha that you found in your report.

[5 points]

0.01

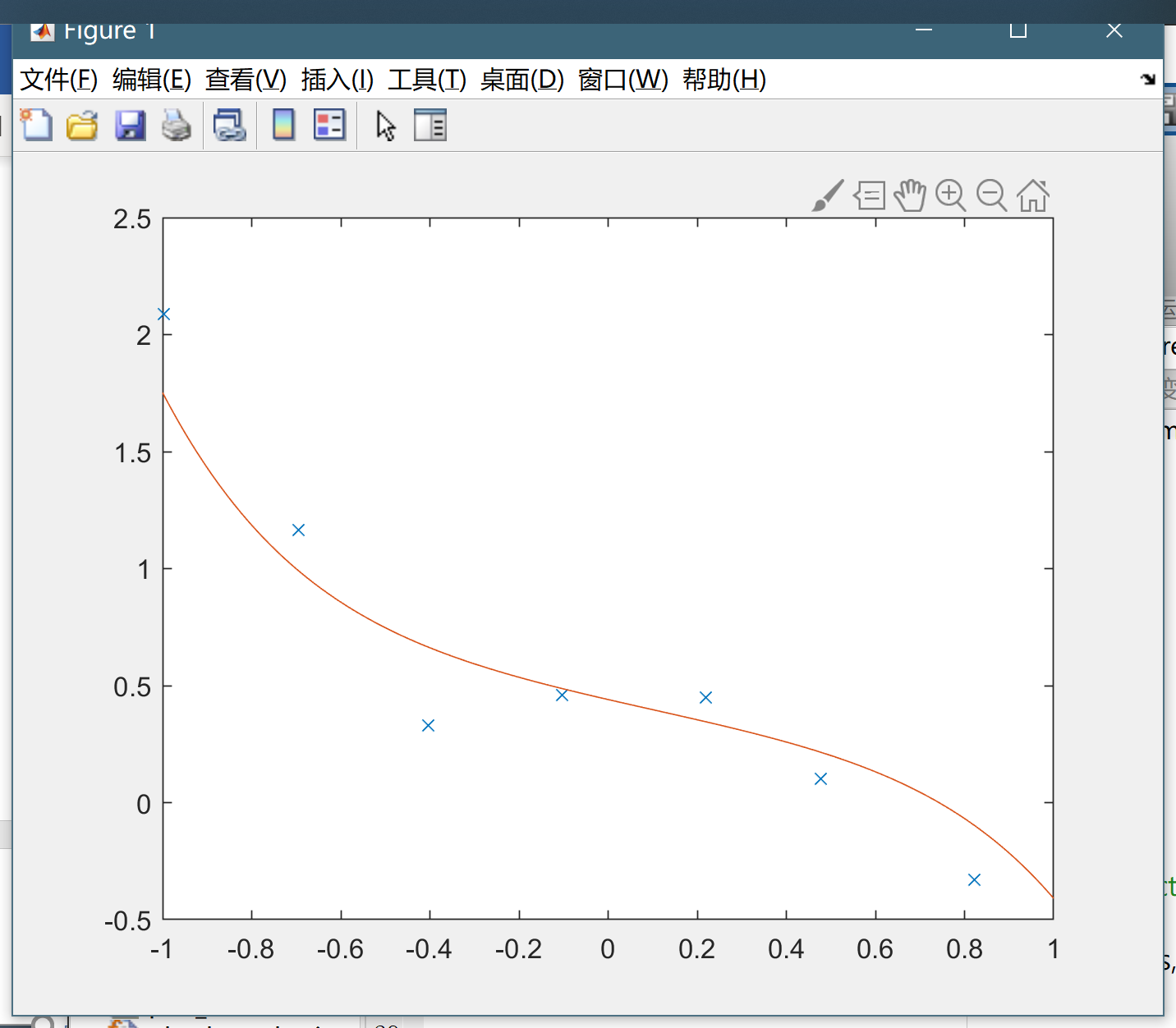
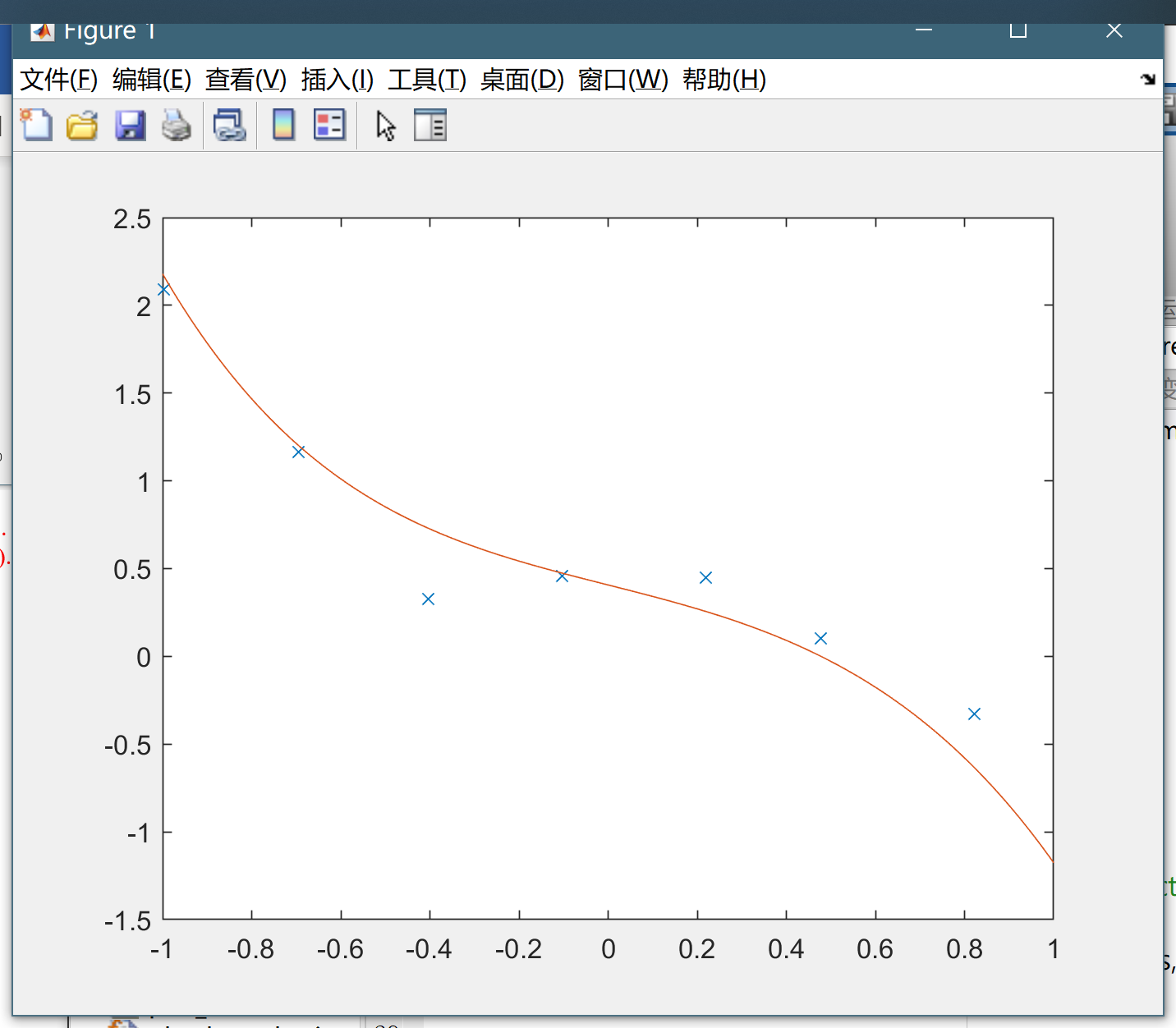
0.1 使用此数

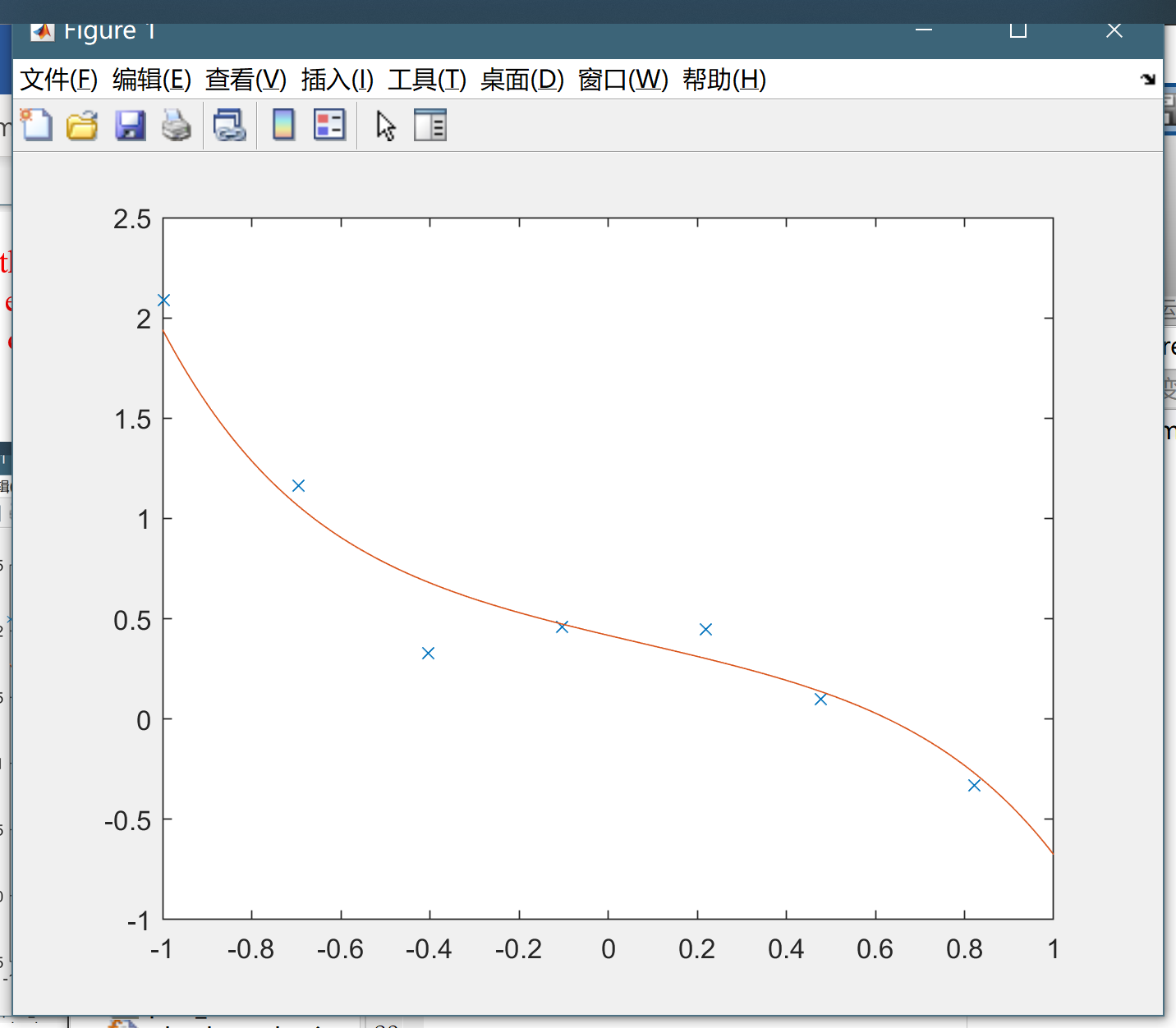
1



Next, experiment with different values of 𝜆 and see how this affects the shape of the hypothesis. Note that gradient\_descent will have to be modified to take an extra parameter, l (which represents 𝜆). Include in your report the plots for a few different values of 𝜆 and comment.

[5 points]





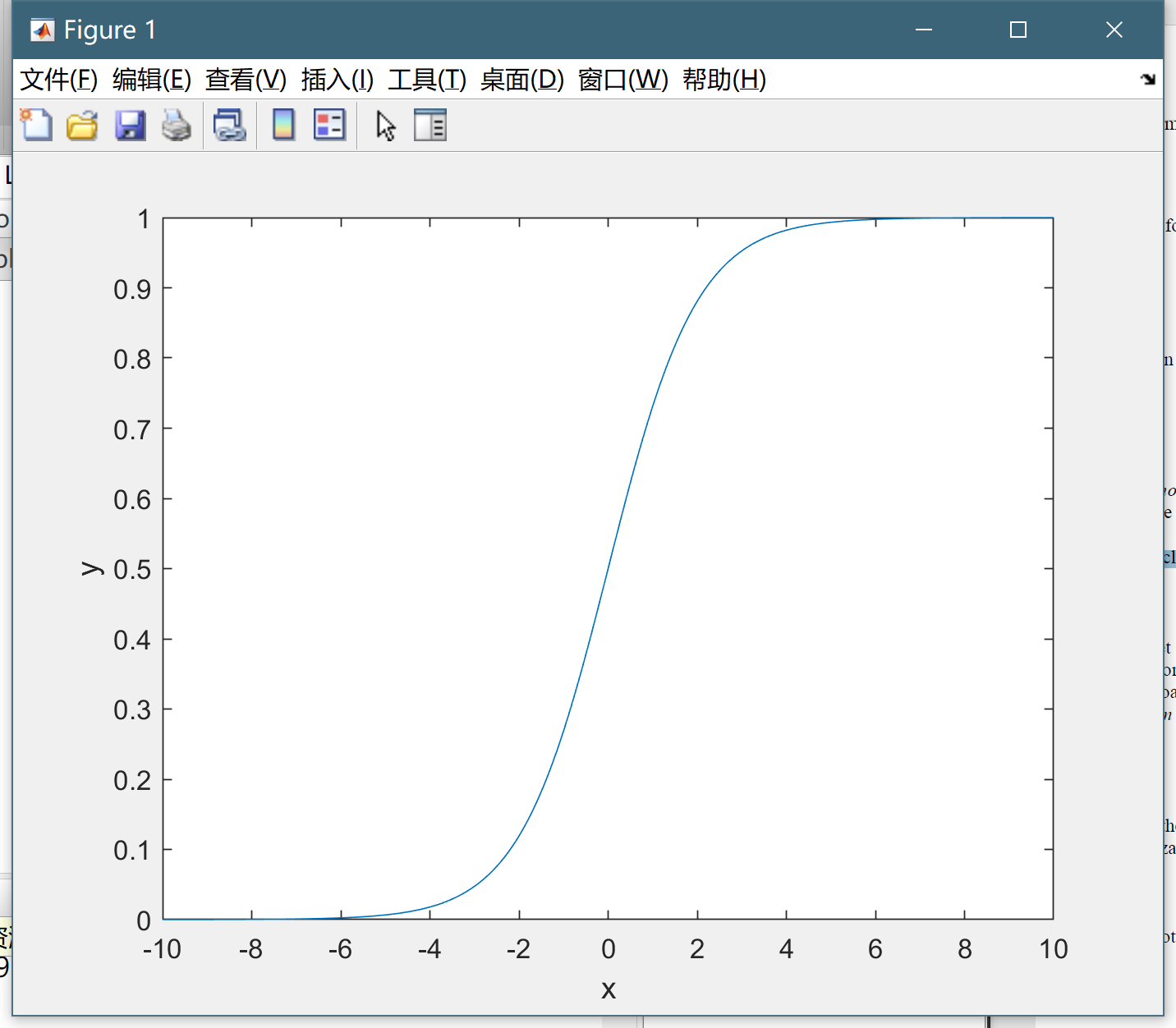
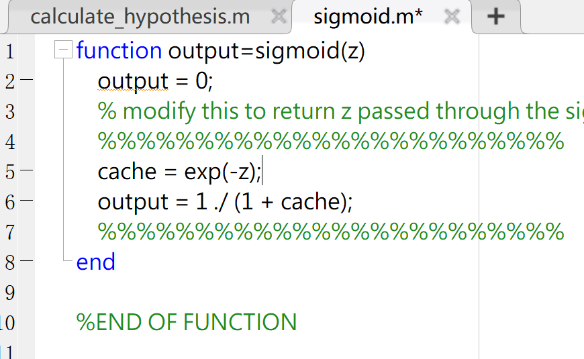
0，2，1

# Ass1 (part2)

## 1. Logistic Regression

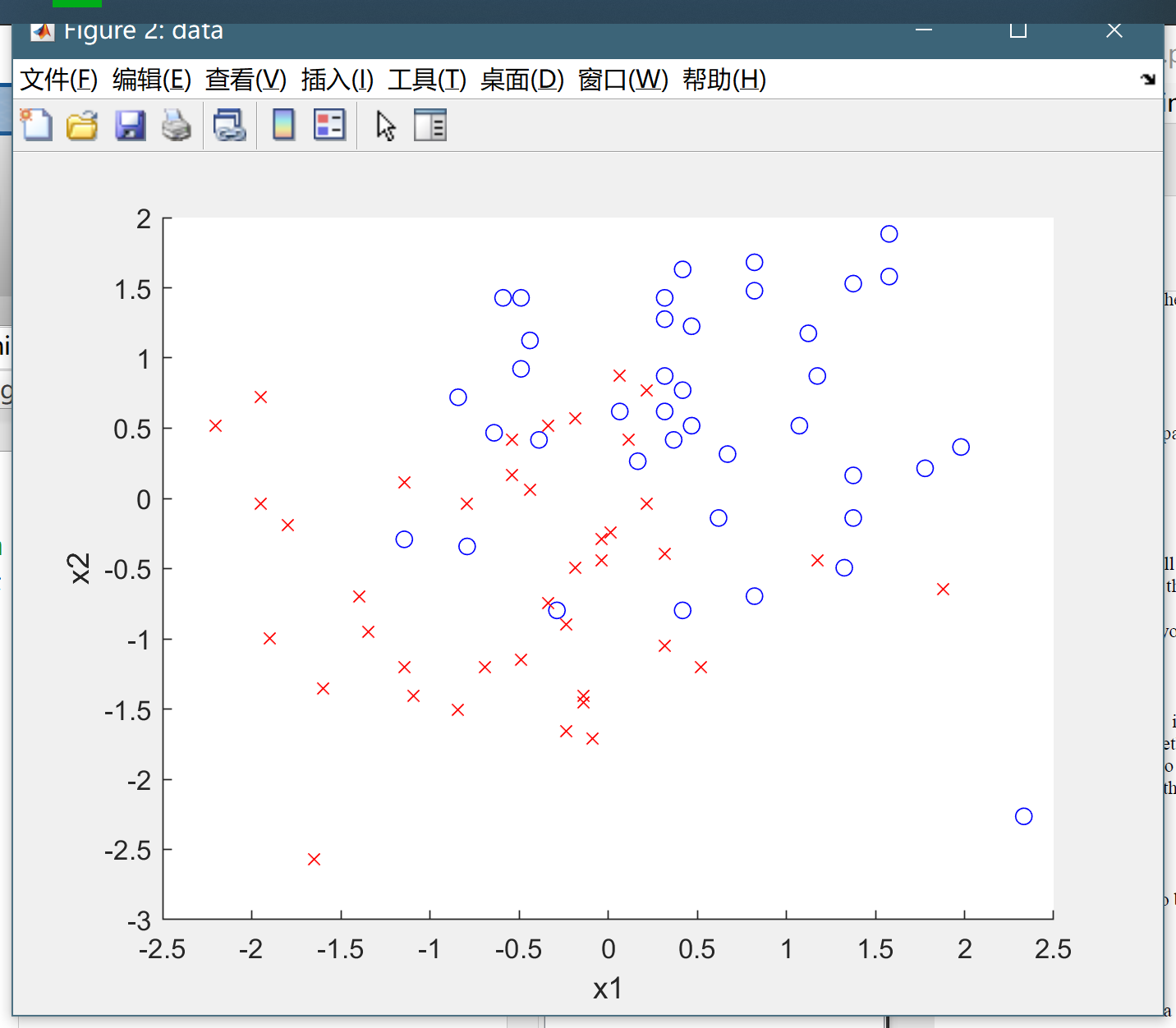
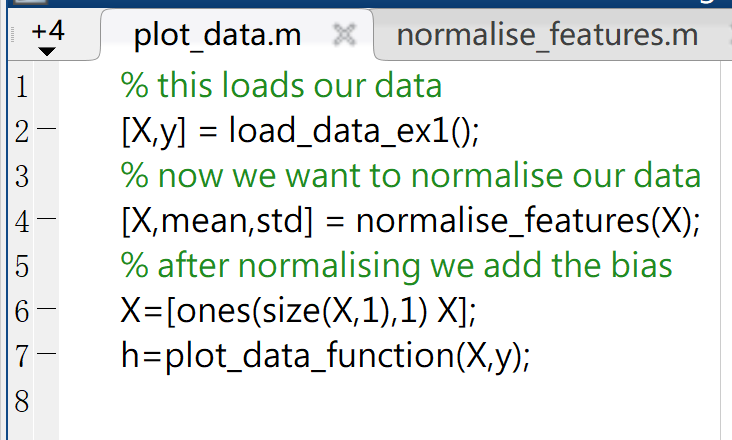
**Task1:** Include in your report the relevant lines of code and the result of the running the plot\_sigmoid\_function.m.

[3 points]



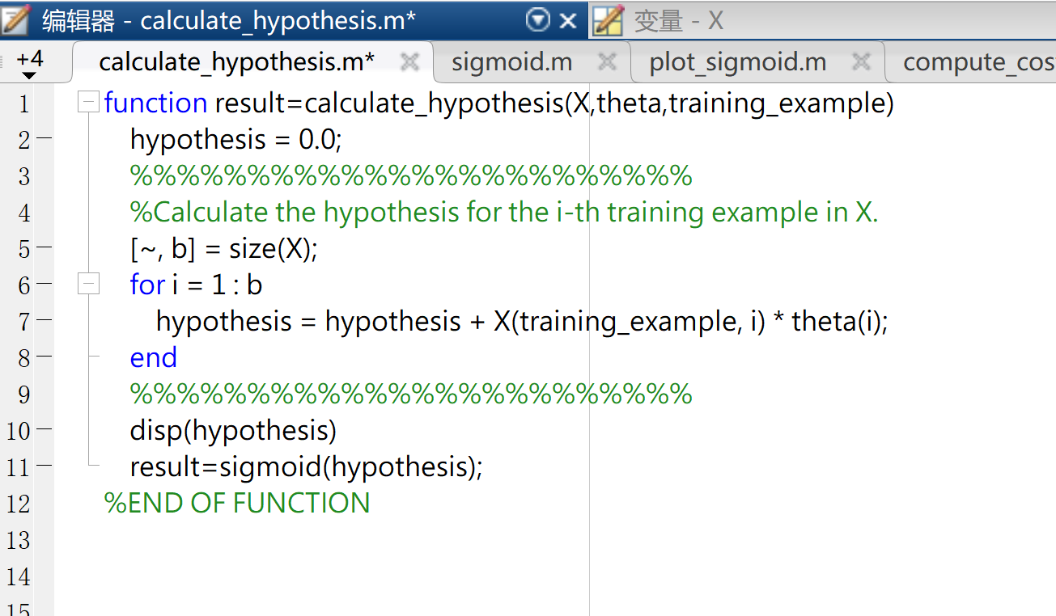
**Task 2**. Plot the data again to see what it looks like in this new format. Enclose this in your report.

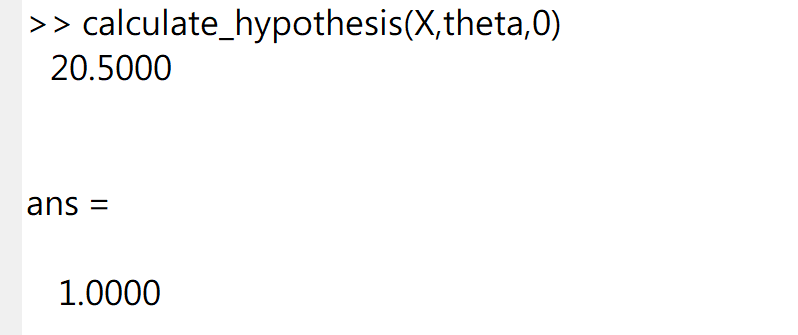
[2 points]



**Task 3.** Modify the *calculate\_hypothesis.m.* The function should be able to handle datasets of any size. Enclose in your report the relevant lines of code.

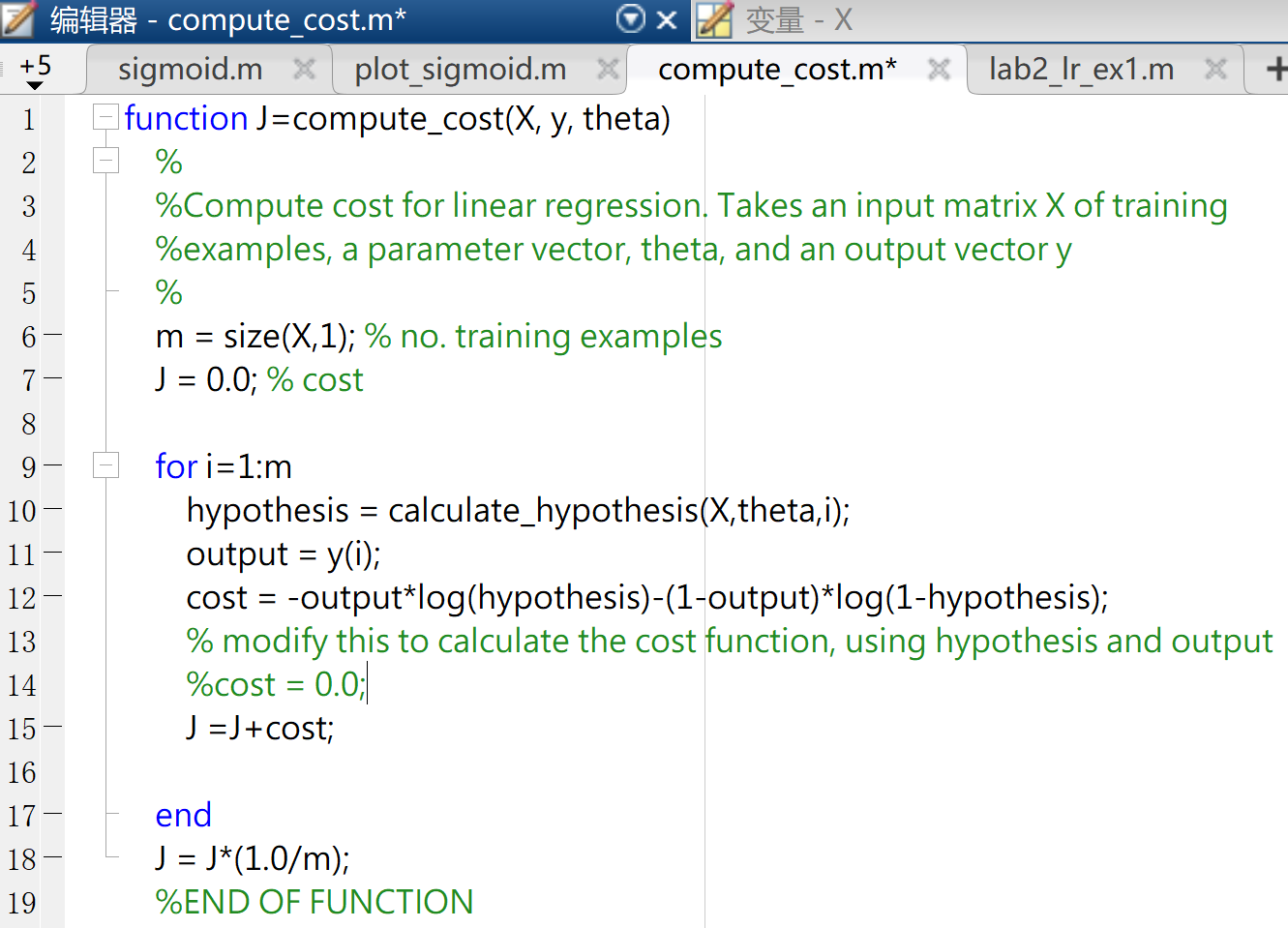
[5 points]

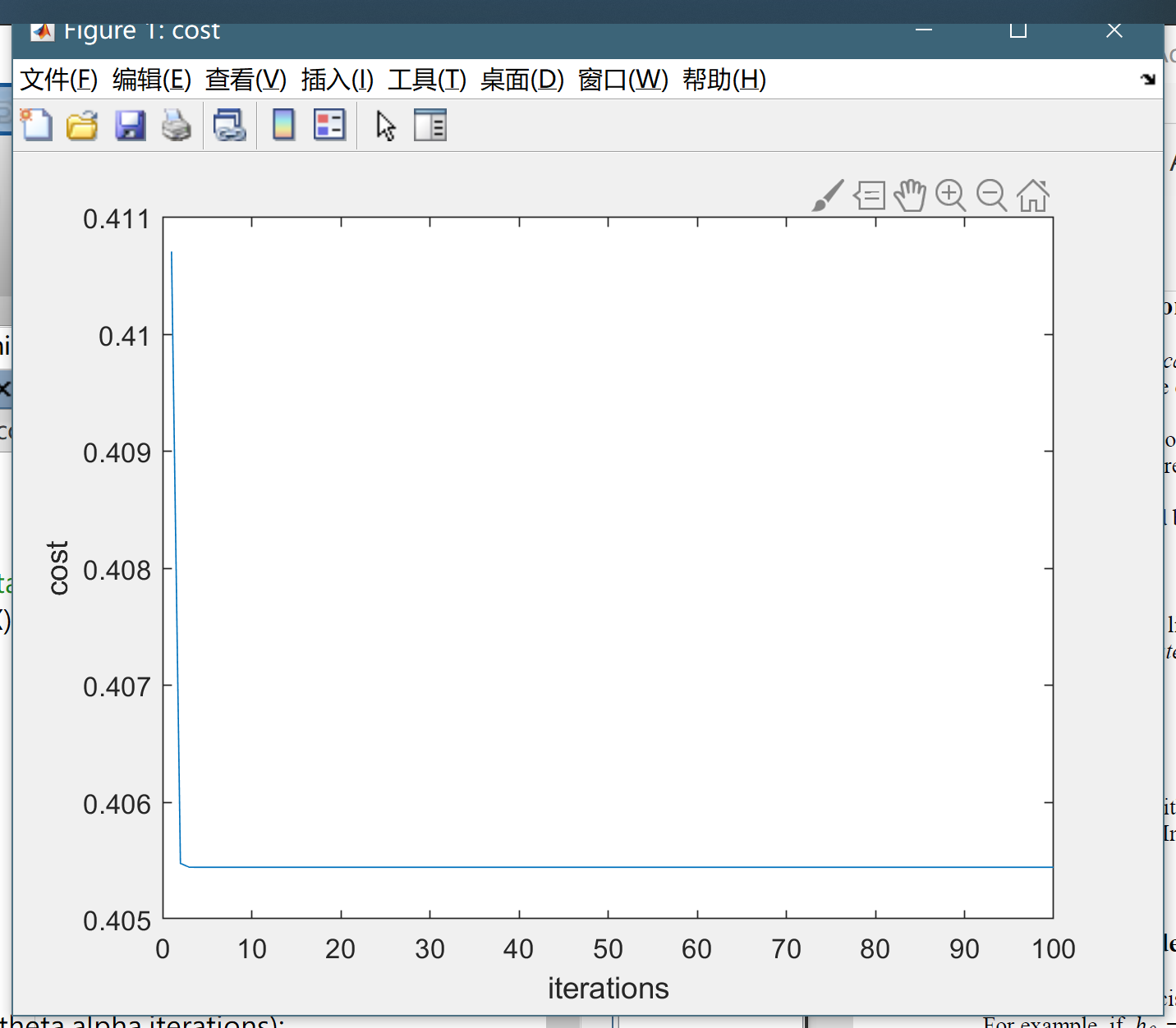
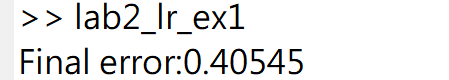




**Task 4**. Modify the line. To calculate a logarithm, you can use log(x). Now run the file *lab2\_lr\_ex1.m* What is the final cost found by the gradient descent algorithm? In your report include the modified code and the cost graph.

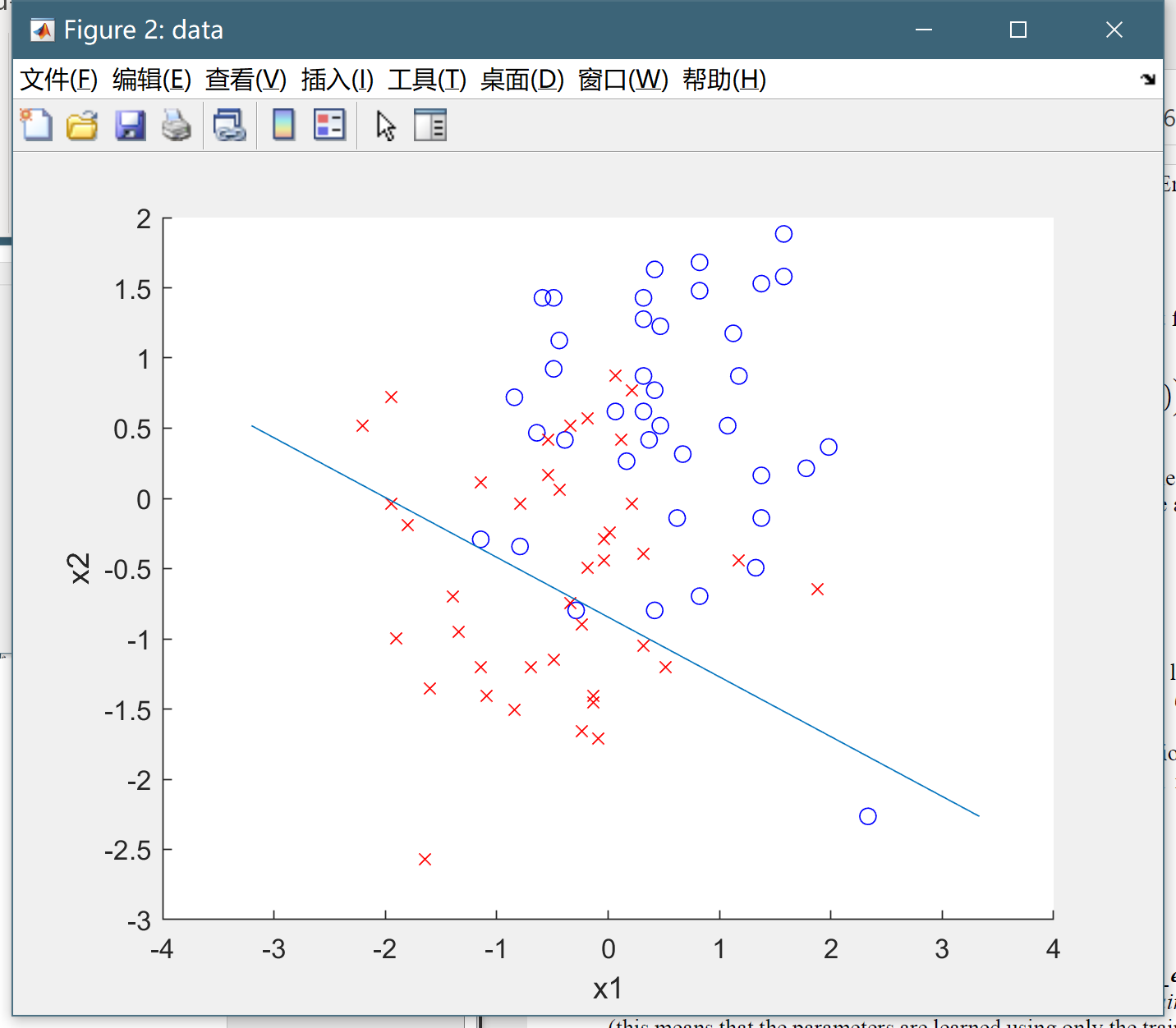
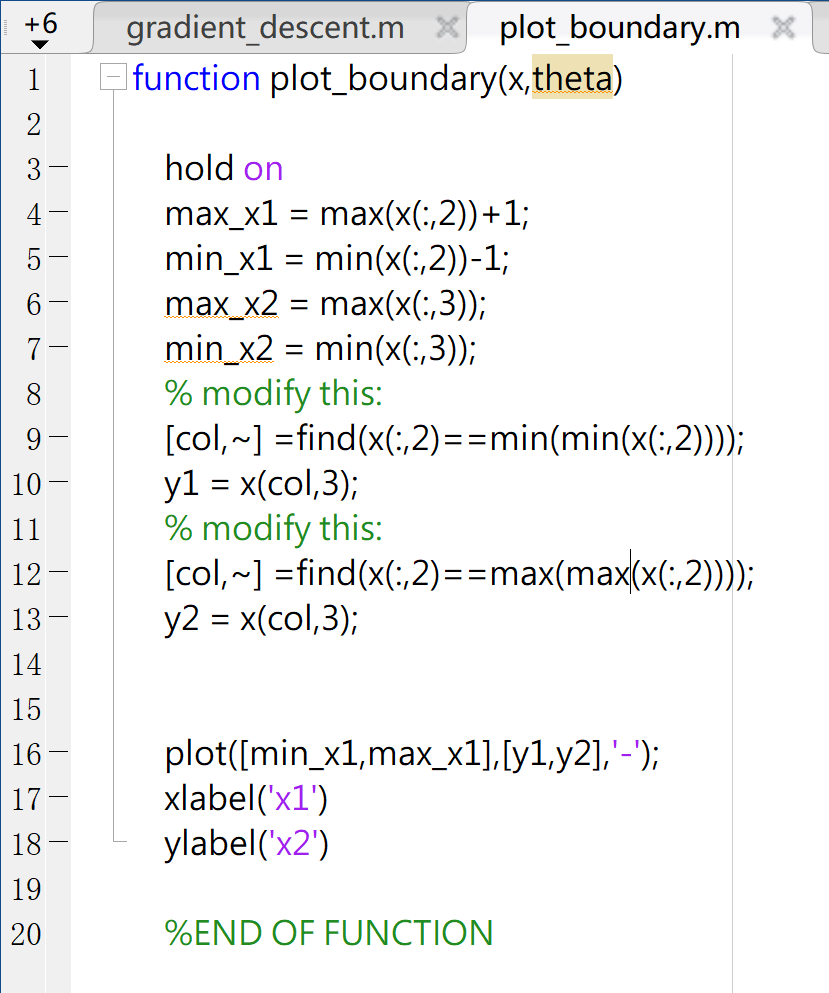
[5 points]





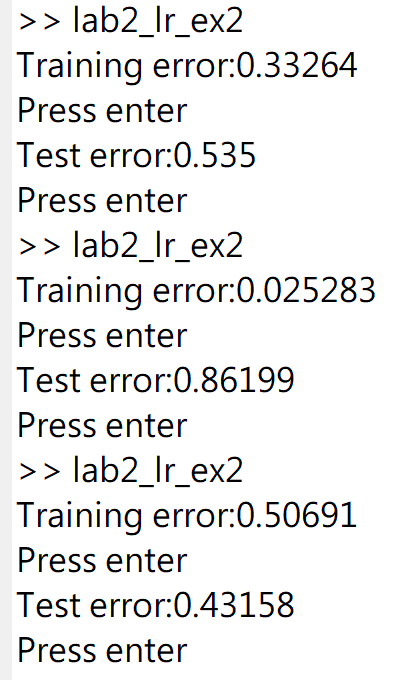
**Task 5.** Plot the decision boundary. Uncomment the relevant plot function in ***lab2\_lr\_ex1.m*** and include the graph in your report.

[5 points]



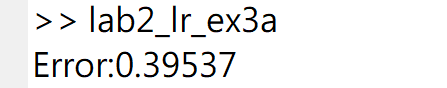
**Task 6.** Run the code in ***lab2\_lr\_ex2.m*** several times. What is the general difference between the training and test error? When does the training set generalize well? Demonstrate two splits with good and bad generalisation and put both graphs in your report.

[2 points]



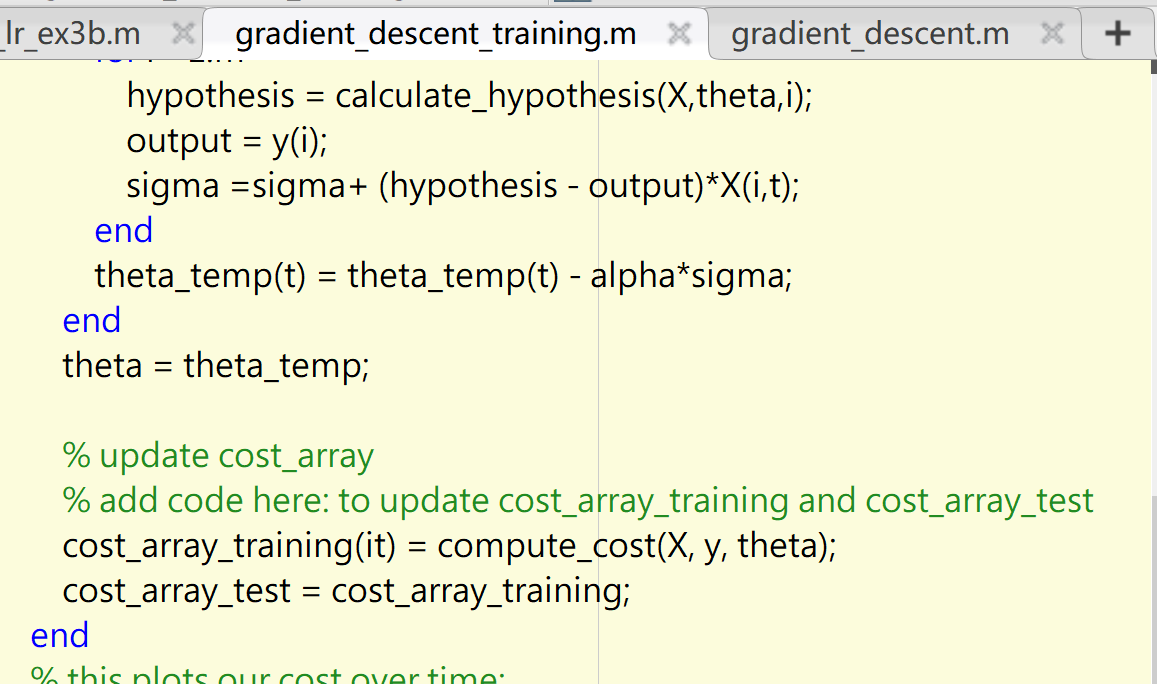
**Task 7.** Run logistic regression on this dataset. How does the error compare to using the original features (i.e. the error found in Task 4)? Include in your report the error and an explanation on what happens.

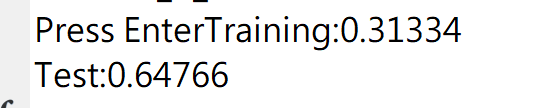
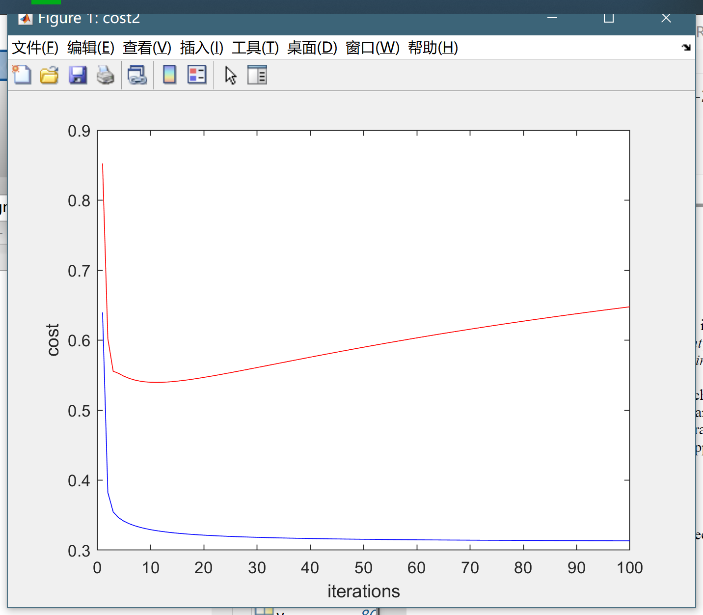
[5 points]

(因为是线性加入？)

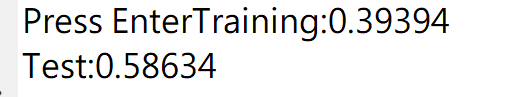
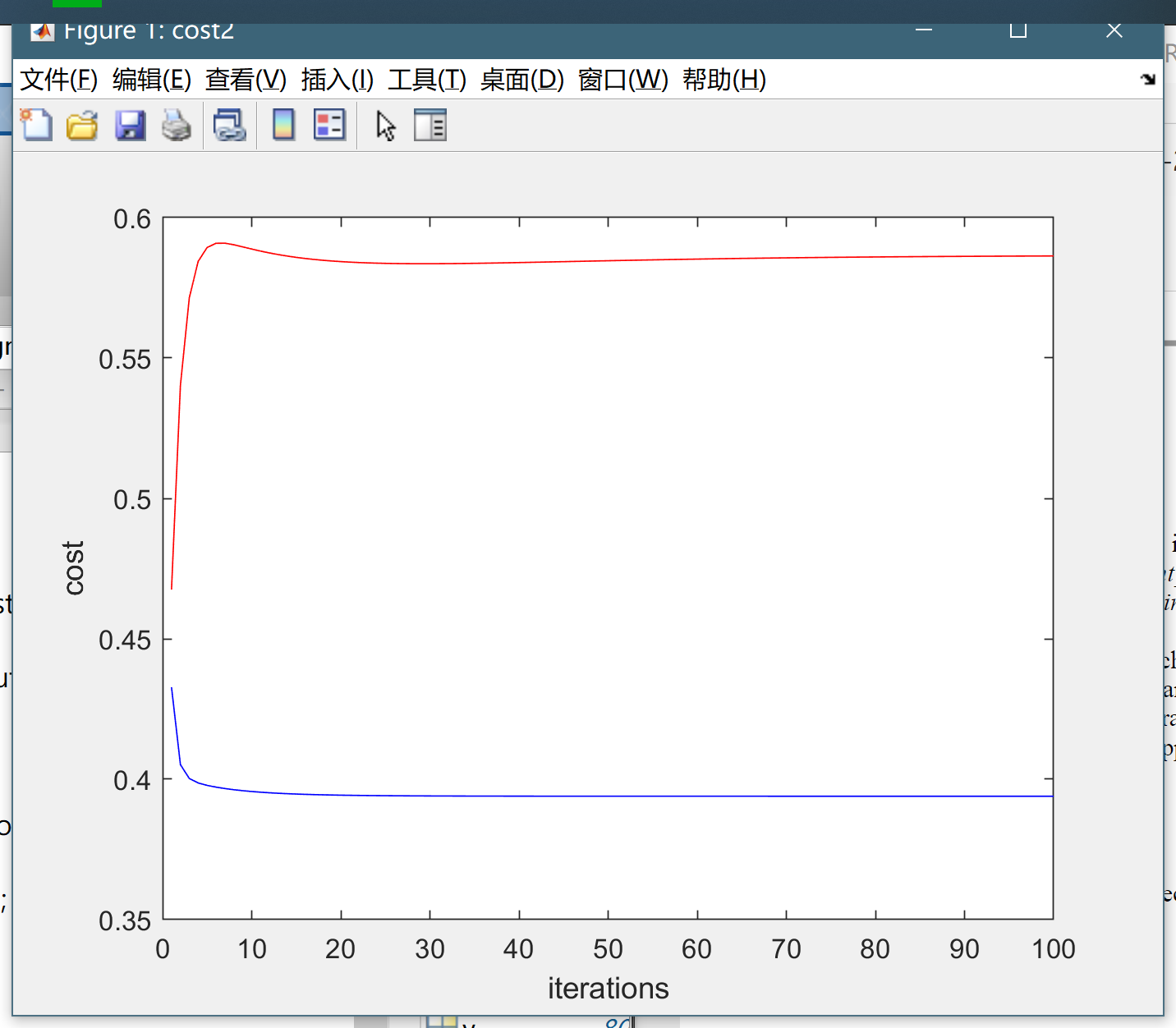
**Task 8**. Add extra features (e.g. a third order polynomial) and analyse the effect. What happens when the cost function of the training set goes down but that of the test set goes up?

[5 points]

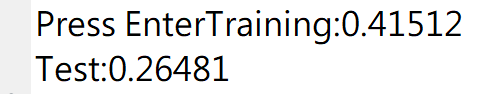
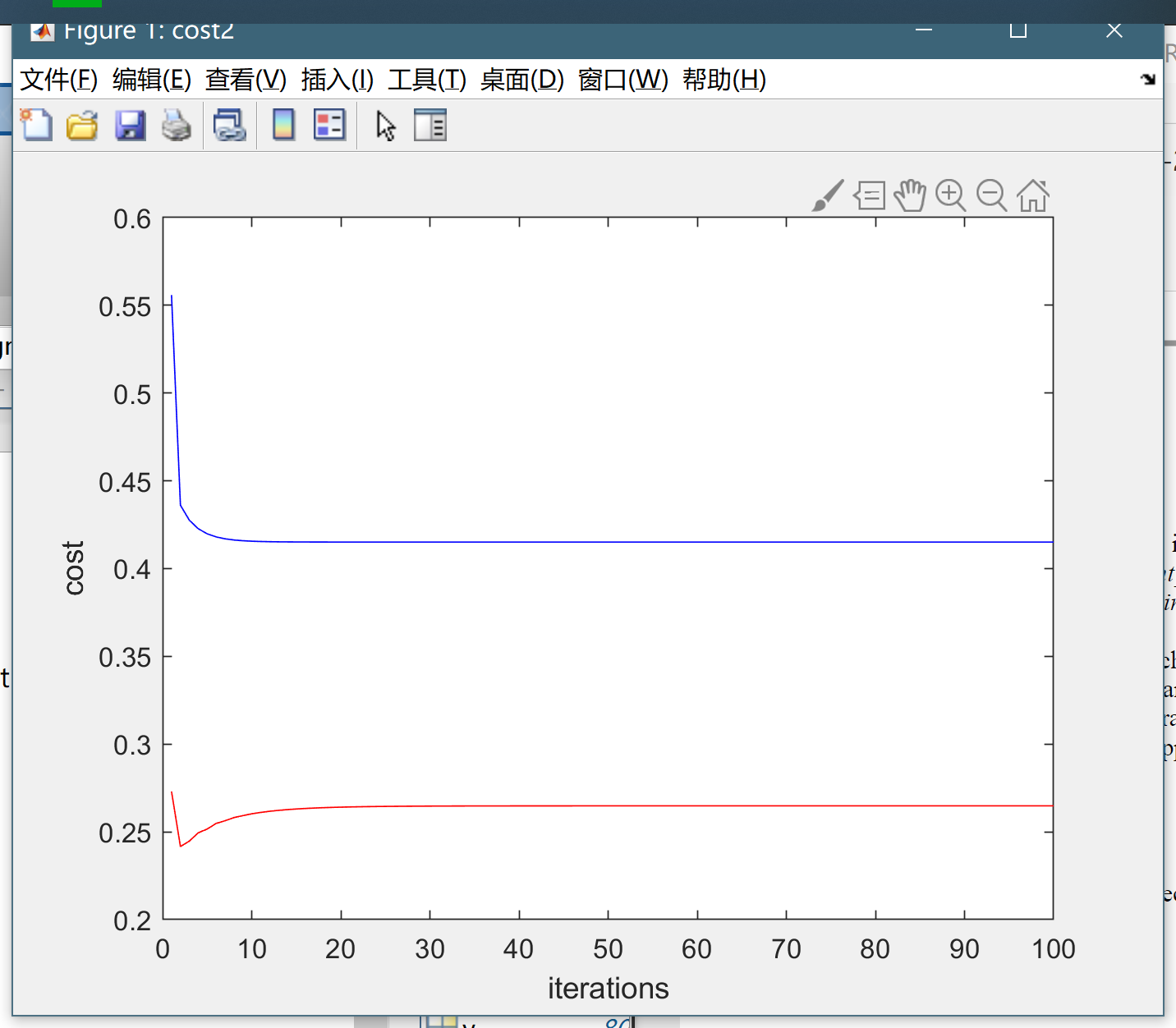




20 training



40 training



70 training

**Task 9.** With the aid of a diagram of the decision space, explain why a logistic regression unit cannot solve the XOR classification problem.

[3 points]

十字型decision space

It can be referred that the decision space of XOR classification is divided by two decision boundaries. However, one logistic regression process could only decide one continuous decision boundary. As a result, it’s impossible for it to solve XOR problem.

## 2. Neural Network