



# Discussion Forums

Get help and discuss course material with the community.

## THIS WEEK'S FORUM

### Week 8

Discuss and ask questions about Week 8.

1907 threads · Last post 6 hours ago

[Go to forum](#)

Forums

All Threads

Search



← All Course Discussions



## ex5 tutorial linearRegCostFunction



Tom Mosher · Mentor · Week 6 · 3 years ago · Edited

Here is a brief tutorial for the `linearRegCostFunction()`.

We last did a linear regression exercise back in ex1, so start with these two tutorials for `computeCost()` and `gradientDescent()`. Since they're vectorized, they work equally well for any multiple-variable linear regression.

[computeCost tutorial](#)

[gradientDescent tutorial](#)

You only need the first three steps of the `gradientDescent()` tutorial, plus scaling by  $1/m$  (ignore the 'alpha' variable, it is not used in this exercise). That's gives us the gradient. Since we let `fmincg()` perform gradient descent for us, we just have



to compute the cost and gradient. We don't use a for-loop over the number of iterations, or use any learning rate. The `minimize` function does that for us.



So now you've got unregularized cost `J`, and unregularized gradient `'grad'`.

For the cost regularization:

- Set `theta(1)` to 0.
- Compute the sum of all of the `theta` values squared. One handy way to do this is `sum(theta.^2)`. Since `theta(1)` has been forced to zero, it doesn't add to the regularization term.
- Now scale this value by `lambda / (2*m)`, and add it to the unregularized cost.

For the gradient regularization:

- The regularized gradient term is `theta` scaled by `(lambda / m)`. Again, since `theta(1)` has been set to zero, it does not contribute to the regularization term.
- Add this vector to the unregularized portion.

That's it. Here is a test case for this function:

<https://www.coursera.org/learn/machine-learning/discussions/O25D0QyKEeWZSyAC5bWQg>

=====

keywords: tutorial linearregcostfunction

↑ 11 Upvotes    Reply    Follow this discussion

This thread is closed. You cannot add any more responses.

**Earliest**

**Top**

**Most Recent**



Tom Mosher · Mentor · 2 years ago



This thread is closed to new comments. If you have a question, please start a new thread.

↑ 0 Upvotes    Reply

MV

Marc de Vernon · 2 years ago



I am having trouble with the cost function. I copied and pasted my solution from `ex1.m`, which passed the grader, and I rewrote the `ex1` version into vectorized. I also manually put the data `X` and `y` into excel and computed the



formula in the PDF. In each case, I get the same answer, which is wrong per the grader.



Any ideas?

The error in the ex5.m output:

Cost at theta = [1 ; 1]: 303.951526

(this value should be about 303.993192)

↑ 0 Upvotes

Hide 1 Reply



Tom Mosher · Mentor · 2 years ago



That's not the test case that the submit grader uses.

Try the tutorials and additional test cases form the Resources menu.

I am currently out of town with a very bad internet connection.

If you need assistance, please start a new thread so that another mentor will pick it up.

↑ 0 Upvotes

ND Ngozi Dozie · 2 years ago



Hi I get the right cost and gradient values but when I try and submit I get the following error:

Program paused. Press enter to continue.

error: linearRegCostFunction: operator +: nonconformant arguments (op1 is 9x1, op2 is 2x1)

error: called from:

error: /Users/ngozidozie/MLX/MLX1/ex5/linearRegCostFunction.m at line 46, column 6

error: /Users/ngozidozie/MLX/MLX1/ex5/fmincg.m at line 77, column 9

error: /Users/ngozidozie/MLX/MLX1/ex5/trainLinearReg.m at line 19, column 7

error: /Users/ngozidozie/MLX/MLX1/ex5/ex5.m at line 168, column 8



When I submit I get the following error:!! Submission failed: unexpected error:  
operator +: nonconformant arguments (op1 is 3x1,op2 is 2x1)

Don't have any idea why this is the case.

↑ 0 Upvotes

Hide 10 Replies



Tom Mosher · Mentor · 2 years ago



The error messages tell the tale. Your code is trying to combine two variables that do not have compatible sizes.

Your code does not work correctly on the data set that the submit grader is using.

Be sure that your code doesn't make any assumptions about the size of the data set.

↑ 0 Upvotes

ZH

ziqu hu · 2 years ago



Hi i get the correct answer for both the ex5 and testcase. but somehow can't get the points for the cost function. what should i do

↑ 0 Upvotes



Tom Mosher · Mentor · 2 years ago



Please post your results from the additional test cases.

↑ 0 Upvotes

ZH

ziqu hu · 2 years ago



Thanks for the reply

```
>> X = [[1 1 1]' magic(3)];
```

```
y = [7 6 5]';
```

```
theta = [0.1 0.2 0.3 0.4]';
```

```
[J g] = linearRegCostFunction(X, y, theta, 0)
```

J =



1.3533

g =

-1.4000

-8.7333

-4.3333

-7.9333

```
>> X = [[1 1 1]' magic(3)];
```

```
y = [7 6 5]';
```

```
theta = [0.1 0.2 0.3 0.4]';
```

```
[J g] = linearRegCostFunction(X, y, theta, 7)
```

J =

5.3763

g =

-1.4000

-8.2667

-3.6333



↑ 0 Upvotes



Tom Mosher Mentor · 2 years ago



Your cost value for the regularized case is incorrect.

↑ 0 Upvotes



Tom Mosher Mentor · 2 years ago



The test case thread says it should be 1.6917.

↑ 0 Upvotes

ZH

ziqu hu · 2 years ago



Thanks Tom, I figured out that I made a very stupid mistake within the cost function.

↑ 0 Upvotes



Karthikeyan Ramakrishnan Mentor · 2 years ago



Hi Tom, for the below example am getting a J value as >

```
X = [[1 1 1]' magic(3)];
```

```
y = [7 6 5]';
```

```
theta = [0.1 0.2 0.3 0.4]';
```

```
[J g] = linearRegCostFunction(X, y, theta, 7)
```

J =3.2783

for the same data set, could you please pass on value of sum of err sq & sum of theta sq (excluding theta(1)), I will just verify where it went wrong.

↑ 0 Upvotes





Tom Mosher · Mentor · 2 years ago · Edited



Here are the values of  $h$  and  $y$ . You already have  $\theta$  and  $\lambda$ . I think you can figure it by hand from there:

```
1  h =  
2  
3      4.4000  
4      5.0000  
5      4.4000  
6  
7  debug> y  
8  y =  
9  
10     7  
11     6  
12     5
```

↑ 0 Upvotes



Tom Mosher · Mentor · 2 years ago

The unregularized part of cost is 1.3533.

The regularized part of the cost is 0.33833.

↑ 0 Upvotes



Ravi Teja · 2 years ago

I have implemented both cost and gradient computation properly and am getting the correct answer. However, when i submit, I'm getting 0 points. I have tried the other test cases and have got them correct. What should I do?

↑ 0 Upvotes

Hide 1 Reply



Rasmus Ranmar · 2 years ago

Hi Ravi

I had the samme problem. I moved some paranthesis around my sum function and submitted and I got my points. Don't know why - it gave samme result when testing the testcases.

↑ 0 Upvotes

GS

Gurkaran Singh · 3 years ago

I think I'm missing an important conceptual piece. Why do I need to use the sum function in my unregularized cost function, but not in the unregularized gradient calculation?

↑ 0 Upvotes

Hide 2 Replies

coursera



Tom Mosher · Mentor · 3 years ago



Use `sum()` when you use element-wise multiplication.

When you use a vector multiplication, the sum is included automatically.

Both methods work. Examples:

```
1 v = [1 ; 2 ; 3]
2 sum(v.^2)
3 v'*v
```

↑ 2 Upvotes

GS

Gurkaran Singh · 3 years ago



Ah, that makes sense. Thanks for the clarification!

↑ 1 Upvote

SC

Sebastian Rodriguez Colina · 3 years ago



Hi, is `h` still the sigmoid function applied to `theta`?

I ask this because I do not find the sigmoid function in the downloaded files

Thanks a lot!

↑ 1 Upvote

Hide 2 Replies



Tom Mosher · Mentor · 3 years ago



No.

Exercise 5 is linear regression. `sigmoid()` is not used.

↑ 1 Upvote

SC

Sebastian Rodriguez Colina · 3 years ago



Thanks, your No remind me that I am suppose to use it for logistic regrestion only.

↑ 1 Upvote





JL

Jing Li · 3 years ago

coursera



Hello! I completed ex1 fine and used it as a reference when coding for ex5. I double checked my code but still can't figure out why my cost and gradient are both just slightly off. Any idea or advice would be greatly appreciated!

0 Upvotes

Hide 1 Reply



Tom Mosher · Mentor · 3 years ago

Perhaps compare your code with what is in the tutorial for this exercise:

<https://www.coursera.org/learn/machine-learning/discussions/m0ZdyjSrEeWddilAC9pDDA>

0 Upvotes



Kumuda Gururao · 3 years ago

For ex5, assignment, i got the cost function to be 303.993, the correct value. But when I wrote the code for gradient, obtained an error saying that 'Index exceeds matrix dimensions'. I did as suggested by you. Don't know where the error lies.

Thanks in advance!

1 Upvote

Hide 26 Replies

[See earlier replies](#)

GK · Gautam Karmakar · 3 years ago

Thanks for reply. But I am still not sure. we use fmincg() to get optimum theta which needs a cost function as parameter. Do I really need this? or can use previous method of using vectors to calculate regularized cost and grad.

0 Upvotes

GK · Gautam Karmakar · 3 years ago



Also tutorial says we only need first three steps of gradientCost function including scale by 1/n. Does it use alpha which tutorial says we don't need to as we are using fmincg()?



↑ 0 Upvotes



Tom Mosher · Mentor · 3 years ago



The vector calculations for cost and grad are what is inside your cost function.

We stopped using a fixed learning rate "alpha" after the first programming exercise. fminunc() in ex2 and fmincg() in ex3, ex4, and ex5 handle managing the learning rate for us.

↑ 0 Upvotes

GK

Gautam Karmakar · 3 years ago · Edited



Thanks Tom. I am not yet successful to complete linearRegCostFunction.m after almost sitting all day since 8.

I am getting cost in two sets of 15 iterations of value 1.15e-01 but grader isn't giving me any marks.

This is what I have done,

created a new CostFunction to calculate non regularized linear regression cost and grad based on X, theta.

Then called fmincg() function with that cost function, initial theta value (initialized to zero) and options.

get J and grad (optimized)

use to add regularized part using supplied lambda.

Note: I didn't use gradientDescent.m (its using computeCost function anyway) as told in the tutorial rather used computeCost function to calculate cost and grad using X, theta.

↑ 0 Upvotes



Tom Mosher · Mentor · 3 years ago



Are you working on assignment ex5?



You should be modifying the script templates that were provided in the ex5 zip archive file - not creating one from scratch.

coursera



↑ 0 Upvotes

GK Gautam Karmakar · 3 years ago



I thought so too but fmincg confused me because it requires a function as parameter. `fmincg(costfunction(), initial_theta,options)` so I thought to create a cost function and put the code to calculate non regularized J and grad calculation inside based on X, theta available. I am sure I am doing wrong but not sure how I handle `fmincg()` without a function written as I see example before in ex3 and ex4 for `fmincg`.

↑ 0 Upvotes



Tom Mosher Mentor · 3 years ago · Edited



That use of "costFunction" is done using the "anonymous function" method. It is really more of a function pointer than a real function.

In ex5, the function "trainLinearReg()" handles this for you.

Open up `trainLinearReg.m` in a text editor, and look at lines 13 through 19. You don't have to change anything here. `trainLinearReg()` calls `fmincg()` using your cost function.

Your `learningCurve()` and `validationCurve()` functions just call `trainLinearReg()` to get the job done.

↑ 0 Upvotes

GK Gautam Karmakar · 3 years ago



Okay, so I was thinking wrong then. in `LinearRegCostFunction.m` all I have to do is write a code to compute J and grad for X, theta. Now grad needs alpha but as later we will be using `fmincg()` we can use `alpha = 1` (means no alpha in the equation?).

grad = alpha/m scaled by sum(X product error).

↑ 0 Upvotes



Tom Mosher Mentor · 3 years ago · Edited



No.



Stop talking about alpha.



There is no alpha in your cost function. There hasn't been since Week 2.

From ex5.pdf:

Here is the cost equation you need to implement:

$$J(\theta) = \frac{1}{2m} \left( \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \right) + \frac{\lambda}{2m} \left( \sum_{j=1}^n \theta_j^2 \right),$$

where  $\lambda$  is a regularization parameter which controls the degree of r

Here is the gradient equation you need to implement:

### 1.3 Regularized linear regression gradient

Correspondingly, the partial derivative of regularized linear regression's cost for  $\theta_j$  is defined as

$$\begin{aligned} \frac{\partial J(\theta)}{\partial \theta_0} &= \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} & \text{for } j = 0 \\ \frac{\partial J(\theta)}{\partial \theta_j} &= \left( \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \right) + \frac{\lambda}{m} \theta_j & \text{for } j \geq 1 \end{aligned}$$

There's no "alpha" variable in any of those.

The learning rate is buried away and hidden inside fmincg(). You don't need to worry about it. You can't see it, and you can't change it even if you wanted to.

↑ 2 Upvotes

GK

Gautam Karmakar · 3 years ago



Thanks a lot Tom, I am progressing now.

↑ 0 Upvotes

E

Ezi · 3 years ago



Hi Tom, I'm really struggling with this assignment, and my frustration has only confused me more. Let me try and stick to this simple test case that you so kindly provided. Here are the results of the test case with lambda = 0:



```
>> X = [[1 1 1]' magic(3)];
```

```
y = [7 6 5]';
```

```
theta = [0.1 0.2 0.3 0.4]';
```

```
[J g] = linearRegCostFunction(X, y, theta, 0)
```

```
J =
```

```
1.3533
```

```
g =
```

```
0
```

```
0
```

```
0
```

```
0
```

Now let me summarize what I've done:

1.  $h$  is product of  $X$  and  $\theta$
2. error is  $h$  minus  $y$
3. I square the error using  $.^2$
4. My unregulated  $J$  is then 1 divided by (2 times  $m$ ) times  $\text{sum}(\text{my squared error})$
5. My unregulated gradient is then  $(\lambda \text{ time } (1 \text{ divided by } m)) \text{ times } (\text{inverse } X \text{ times my error})$
6. I then set  $\theta(1)$  to 0 using the method  $(:,1) = 0$
7. I determine my regulated  $J$  as  $(\lambda \text{ divided by } (\text{double } m)) \text{ times the sum of } (\theta \text{ squared}) - \text{again I square using } .^2$



8. J now become my unregulated J plus my regulated J

9. Then I determine my regulated grad as  $(\lambda \text{ divided by } m)$  times theta

10. grad now becomes my unregulated grad plus my regulated grad

I have to say my step 5 above is where I'm most unsure/confused; but add in my frustration and I'm not sure of anything now (lol).

Please please can you tell me where I've fallen off nudge me ever so cleverly as you do back to a joyful outcome?

↑ 0 Upvotes



Tom Mosher · Mentor · 3 years ago · Edited



Sorry, I'm off-duty this weekend. If you want another mentor to spot your question, please start a new thread.

Or someone else from the community can provide help.

↑ 0 Upvotes

SC

Sebastian Rodriguez Colina · 3 years ago



I think I found your problem it is indeed step 5, why do you multiply with inverse of X? Should be X'. At least that is the only problem I find. Besides that, it seems our steps are the same and it worked for me ;)

↑ 0 Upvotes

JR

Jonathan Roman · 2 years ago · Edited



I'm confused how to apply fmincg to gradient descent above and what parameters and functions to pass to it. Can someone clarify?

↑ 0 Upvotes



Tom Mosher · Mentor · 2 years ago



The trainLinearReg() function calls fmincg() for you. You do not need to do this yourself.

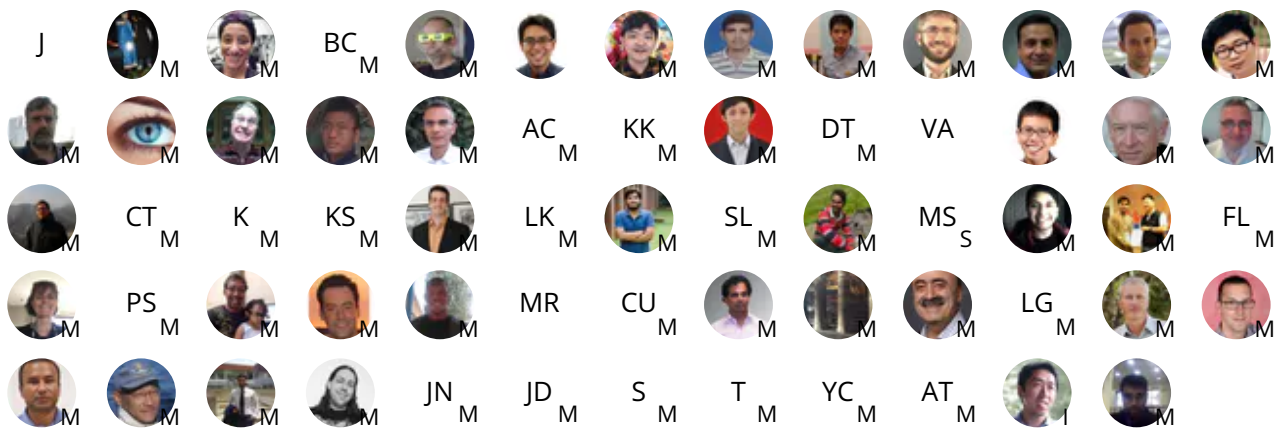
↑ 0 Upvotes



DESCRIPTION

Welcome to the course discussion forums! Ask questions, debate ideas, and find classmates who share your goals. Browse popular threads below or other forums in the sidebar.

MODERATORS



[Learn more about becoming a Mentor](#)

[Forum guidelines](#)

